

**UNIVERSIDADE NOVE DE JULHO
PROGRAMA DE MESTRADO PROFISSIONAL EM ADMINISTRAÇÃO
GESTÃO DE PROJETOS**

**AVALIAÇÃO DE MODELOS MATEMÁTICOS PARA A RESOLUÇÃO DE JOB
SHOP PROBLEM COM A UTILIZAÇÃO DE RECURSOS HUMANOS
ESPECIALISTAS EM PROJETOS**

RENATO PENHA

SÃO PAULO

2012

RENATO PENHA

**AVALIAÇÃO DE MODELOS MATEMÁTICOS PARA A RESOLUÇÃO DE JOB
SHOP PROBLEM COM A UTILIZAÇÃO DE RECURSOS HUMANOS
ESPECIALISTAS EM PROJETOS**

Dissertação apresentada ao Programa de Mestrado Profissional em Administração: Gestão de Projetos da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de **Mestre em Administração**.

Orientadora: Profa. Dra. Cláudia Terezinha Kniess

Co-Orientador: Prof. Ms. Daniel Reed Bergmann

**SÃO PAULO
2012**

RENATO PENHA

**AVALIAÇÃO DE MODELOS MATEMÁTICOS PARA A RESOLUÇÃO DE JOB
SHOP PROBLEM COM A UTILIZAÇÃO DE RECURSOS HUMANOS
ESPECIALISTAS EM PROJETOS**

Banca Examinadora:

Presidente: Profa. Dra. Cláudia Terezinha Kniess, Orientadora - Uninove

Membro: Prof. Ms. Daniel Reed Bergmann - Uninove

Membro: Profa. Dra. Cristina Dai Prá Martens - Uninove

Membro: Prof. Dr. Humberto Gracher Riella - Universidade Federal de Santa Catarina

São Paulo, 26 de junho de 2012.

DEDICATÓRIA

Ao meu pai José Penha, pelo amor e educação.

AGRADECIMENTO

Ao professor Marcello Muniz da Silva, pela dedicação e felicidade na escolha do tema deste trabalho.

Aos meus orientadores, Cláudia Terezinha Kniess e Daniel Bergmann, pelos estudos em conjunto, pela compreensão, profissionalismo e que tornaram possível a conclusão deste trabalho.

A minha mãe Aparecida, pelo amor e incentivo.

A minha esposa Tatiane e meu filho Caio, pelo carinho e paciência durante a ausência no período de elaboração deste trabalho.

A minha turma de mestrado, especialmente ao meu colega de estudos Alexandre de Alcântara Mesquita, pela sapiência, incentivo e apoio durante as fases críticas do curso.

Homenagem póstuma especial ao meu pai José Penha, que sempre comemorou cada vitória alcançada em minha vida.

Finalmente, agradeço aos meus AMIGOS que contribuíram para tudo dar certo no final.

*“Pessoas não fracassam.
Elas simplesmente desistem.”*

Henry Ford

RESUMO

Um projeto pode ser definido como um sistema complexo. Esse exige o emprego de recursos (recursos humanos, materiais, tecnologia, entre outros), alocados entre fins alternativos, como meio de atingir objetivos específicos, mediante a presença de restrições de diferentes ordens. Tradicionalmente, as técnicas de gerenciamento de projetos conferem grande ênfase à definição de atributos associados às atividades (estimativas de recursos, sequenciamento de atividades, obtenção de estimativas de prazos e de custos, elaboração do cronograma e do orçamento do projeto). Já o gerenciamento de portfólio de projetos tem como objetivo estabelecer critérios destinados à priorização de projetos de uma carteira, mas até o presente lida de forma incipiente com o *trade-off* que emerge em razão da (i) escassez de recursos especialistas; (ii) (re)definição de critérios e (iii) avaliação do impacto da reprogramação de um ou mais projetos, a depender do(s) objetivo(s) a serem alcançados. Em relação aos recursos humanos, inclusive recursos especialistas, o planejamento, alocação e priorização destes recursos é feita pela gestão de portfólio de forma única, projeto a projeto. Este tratamento pode causar disputa interna pela utilização do mesmo recurso ou até mesmo a sua subutilização, e pode se agravar em ambientes de desenvolvimento de software devido ao alto grau de interdependência, incerteza e risco de cada projeto. Muitos ambientes de desenvolvimento de softwares são marcados pela escassez de Recursos Humanos Especialistas – RHE. Embora a adoção de técnicas de nivelamento, paralelismo e compressão, auxiliem no planejamento e otimização de cada projeto, a interdependência entre esses demanda a integração de processos de gerenciamento. Tal necessidade está relacionada ao chamado *job shop problem* – JSP. A literatura sugere algumas formas de lidar com esse aspecto. Uma possível solução pode ser dar por meio da introdução de métodos de otimização, envolvendo a minimização do prazo ou do custo total dos projetos da carteira condicionada a restrições de precedência ou de uso de RHE, por exemplo. Como aqui sugerido, tal método pode ser desenvolvido e implantado tendo como base (i) as alternativas técnicas e (ii) as estimativas de recursos que compreendem parâmetros diretamente relacionados à construção do cronograma e do orçamento dos projetos da carteira. Neste contexto, o objetivo deste trabalho é avaliar modelos matemáticos de algoritmo genético e de otimização e suas contribuições para a resolução de *Job Shop Problem* em projetos de desenvolvimento de software com a utilização de recursos humanos especialistas. Essa pesquisa é de caráter exploratório, pois visa integrar aspectos relacionados ao gerenciamento

de prazos, gerenciamento de custos, gerenciamento de riscos tendo em vista a necessidade de alocação de recursos especialistas em um ambiente de multi-projetos. Como resultado, foi possível identificar as vantagens e limitações e apresentar uma análise comparativa entre indicadores de eficácia entre três modelos matemáticos para a solução de JSP marcado pela presença de alocação de recursos humanos especialistas em uma empresa de desenvolvimento de software por meio de um estudo de caso.

Palavras-chave: Modelos matemáticos, recursos humanos especialistas, job shop problem – JSP, projetos de desenvolvimento de software.

ABSTRACT

A project can be defined as a complex system. This requires the use of resources (human, material, technology, etc.), allocated among alternative uses, as a means to achieve specific goals by the presence of constraints of different orders. Traditionally, the techniques of project management attach great emphasis to the definition of attributes associated with the activities (resource estimates, sequencing activities, obtaining estimates of time and cost, schedule and preparation of project budget). Already the project portfolio management aims to establish criteria for the prioritization of projects within a portfolio, but even this deals only incidentally with the trade-off that emerges because of (i) scarcity of experts, (ii) (re) definition of criteria and (iii) assessing the impact of the rescheduling of one or more projects, depending on the (s) objective (s) to be achieved. Regarding human resources, including resource specialists, planning, allocation and prioritization of these resources is made by the portfolio management in a single project to project. This treatment can cause internal strife by using the same resource or even its underuse, and may worsen in software development environments due to the high degree of interdependence, uncertainty and risk of each project. Many software development environments are marked by shortages of Human Resources Specialists - HRS. Although the adoption of smoothing techniques, parallelism and compression, assist in planning and optimization of each project, the interdependence between these demands the integration of management processes. This need is related to the so called job shop problem - JSP. The literature suggests ways to deal with this aspect. A possible solution may be to give through the introduction of optimization methods, involving the minimization of the period or the total cost of the projects in the portfolio subject to precedence constraints or use of RHE, for example. As suggested here, such a method can be developed and deployed based on (i) the technical alternatives and (ii) estimates of resources that include parameters directly related to the construction schedule and budget of the projects in the portfolio. In this context, the objective of this study is to evaluate mathematical models and genetic algorithm optimization and its contributions to solving Job Shop Problem in software development projects using human resources specialists. This research is exploratory, it aims to integrate aspects related to time management, cost management, risk management in view of the need to allocate resources experts in a multi-project environment. As a result, it was possible to identify the advantages and limitations and present a comparative analysis between performance indicators among three mathematical models for

solving JSP marked by the presence of skilled human resource allocation in a software development company through a study of case.

Keywords: Mathematical models, human resources specialists, job shop problem - JSP, software development projects.

LISTA DE FIGURAS

Figura 1 - Ciclo de vida clássico do desenvolvimento de software.....	20
Figura 2 - Modelo em espiral de desenvolvimento de software.....	23
Figura 3 - Os 4 estágios do gerenciamento de projetos de desenvolvimento de software.....	25
Figura 4 - Diferença de conhecimento utilizado entre um profissional especialista e um não especialista.....	33
Figura 5 - Representação aproximada da curva de aprendizado de indivíduos submetidos a treinamento comparado com a de especialistas.....	34
Figura 6 - Desempenho comparativo entre especialistas e não especialistas em um determinado domínio.....	35
Figura 7 - Job shop clássico 3x4.....	37
Figura 8 - Solução ótima para um job shop clássico 3x3.....	37
Figura 9 - Classificação hierarquizada dos modelos de solução para o problema de JSP	39
Figura 10 - Exemplo de algoritmo genético.....	43
Figura 11 - Exemplo de distribuição crossover uniforme.....	61
Figura 12 - Processo de geração de populações consecutivas.....	62
Figura 13 - Algoritmo usado para construção de parâmetros de tempo ativos.....	64
Figura 14 - Algoritmo proposto por Nowicki and Smutnicki (1996).....	65
Figura 15 - Algoritmo usado busca local.....	66
Figura 16 - Algoritmo Busca Tabu modificado.....	69
Figura 17 - Arquitetura do modelo proposto por Müller, Rodrigues e Gómez (2006).....	70

LISTA DE QUADROS

Quadro 1 - Tipos de manutenções de software.....	18
Quadro 2 - Características de sistema e subsistema.....	22
Quadro 3 - Classificação dos profissionais que criam conhecimento.....	32
Quadro 4 - Notação e definição de variáveis associadas ao início, precedências, duração e probabilidade de finalização do projeto e portfólio de projetos.....	55
Quadro 5 - Notação e definição de variáveis associadas as atividades, momento de início e término nos nós, ao início, relação de precedência, duração e probabilidade de finalização do projeto.....	56
Quadro 6 - Notação e definição dos parâmetros de tipo, custo e momento de contração e liberação associados ao emprego de recursos especialistas.....	56
Quadro 7 - Notação e definição dos parâmetros associados à penalidade fixa e variável em função do atraso no k-ésimo projeto do portfólio.....	57
Quadro 8 - Notação e definição do custo total fixo e variável do atraso e emprego de recursos especialistas.....	57
Quadro 9 - Modelos utilizados para resolução de JSP.....	75
Quadro 10 - Indicadores de eficácia do modelo de otimização propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) para a solução de JSP.....	76
Quadro 11 - Quantidade total de projetos e projetos com problemas de desvio de prazos originados por sequenciamento de atividades e escalonamento de recurso.....	80
Quadro 12 - Aspectos relacionados a entrevista realizada com o gestor de projetos de uma empresa desenvolvimento de software.....	81
Quadro 13 - Aderência dos modelos de Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) x indicador de eficácia.....	83
Quadro 14 - Características dos modelos de Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) em relação ao tratamento do JSP.....	89

LISTA DE ABREVIATURAS E SIGLAS

AMP	Ambiente de Múltiplos Projetos
CPM	Critical Path Method
E&P	Exploração e Produção
ERV	Earned Risk Value
GP	Gestão de Projetos
GPP	Gerenciamento de Portfólio de Projetos
GRP	Gerenciamento de Riscos em Projetos
HGA	Hybrid Genetic Algorithm
JSP	Job Shop Problem
NPV	Net Present Value
PE	Planejamento Estratégico
PERT	Program Evaluation and Review Technique
PMI	Project Management Institute
RCPSP	Resource Constraint Project Scheduling Problem
RE	Recurso Especialista
RHE	Recursos Humanos Especialistas
RO	Recursos Ordinários
TI	Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	1
1.1	FUNDAMENTAÇÃO DA PESQUISA	4
1.2	OBJETIVOS	6
1.2.1	OBJETIVO GERAL	6
1.2.2	OBJETIVOS ESPECÍFICOS	7
1.3	ESTRUTURA DO TRABALHO	7
2	REFERENCIAL TEÓRICO	9
2.1	IMPORTÂNCIA DO GERENCIAMENTO DE PROJETOS E TIPOLOGIA DE AMBIENTES	9
2.2	GERENCIAMENTO DE PROJETOS	10
2.2.1	GERENCIAMENTO DE PORTFÓLIO DE PROJETOS	12
2.2.2	GERENCIAMENTO DE RISCOS	14
2.3	CARACTERÍSTICAS DE PROJETOS DE DESENVOLVIMENTO DE SOFTWARES	17
2.4	PROGRAMAÇÃO E ALOCAÇÃO DE RECURSOS EM PROJETOS	29
2.4.1	RECURSOS HUMANOS ESPECIALISTAS	31
2.4.2	CLASSIFICAÇÃO DE RECURSOS HUMANOS ESPECIALISTAS	32
2.4.3	RECURSOS HUMANOS ESPECIALISTAS NO AMBIENTE DE DESENVOLVIMENTO DE SOFTWARE	35
2.5	O MODELO CLÁSSICO DE JOB SHOP PROBLEM	36
3	METODOLOGIA	46
3.1	TIPO DE PESQUISA	46
3.2	PESQUISA QUALITATIVA	46
3.3	ESTRATÉGIA DE PESQUISA	47

3.4	TÉCNICA DE PESQUISA	48
3.5	O ESTUDO DE CASO	48
3.6	A ESCOLHA DO MÉTODO	50
3.7	DELINEAMENTO DA PESQUISA.....	51
3.8	PROCEDIMENTOS METODOLÓGICOS	51
3.9	LIMITAÇÕES DA PESQUISA	53
4	RESULTADOS E DISCUSSÃO	54
4.1	MODELO PROPOSTO POR LASLO (2010)	54
4.1.1	APRESENTAÇÃO DO MODELO.....	54
4.2	MODELO PROPOSTO POR GONÇALVES, MENDES E RESENDE (2004)	60
4.2.1	APRESENTAÇÃO DO MODELO.....	61
4.2.2	PROCESSO DE CONSTRUÇÃO	62
4.2.3	PROCEDIMENTO DE BUSCA LOCAL.....	65
4.3	MODELO PROPOSTO POR MÜLLER, RODRIGUES E GÓMEZ (2006)	67
4.3.1	APRESENTAÇÃO DO MODELO.....	67
4.3.2	ARQUITETURA DO MODELO.....	70
4.4	CONTRIBUIÇÕES DOS MODELOS PARA O SOLUÇÃO DO JSP.....	71
4.5	REQUISITOS DE ADERÊNCIA DOS MODELOS X SOLUÇÃO JSP	76
4.6	AVALIAÇÃO DOS MODELOS EM ESTUDO DE CASO	78
4.6.1	CARACTERIZAÇÃO DA EMPRESA.....	78
4.6.2	CENÁRIO ATUAL DA EMPRESA.....	79
4.6.3	PROBLEMAS ENFRENTADOS PELA EMPRESA.....	79
4.6.4	AVALIAÇÃO DOS MODELOS PARA RESOLUÇÃO DO JSP PARA O ESTUDO DE CASO	81
4.6.5	DISCUSSÃO	86

5	CONCLUSÃO.....	89
----------	-----------------------	-----------

1 INTRODUÇÃO

Organizações são sistemas abertos e complexos que processam recursos como meio de atingir objetivos na presença de múltiplas restrições, internas e externas (TAVARES, 2002). Ao estruturar atividades rotineiras (processos) e ou não rotineiras (projetos), as organizações empregam fatores produtivos escassos entre fins alternativos, mediante adoção de uma dada tecnologia.

Particularmente, o gerenciamento de projetos compreende os esforços destinados a conceber, preparar, organizar, dirigir e controlar certas ações orientadas a alocação de recursos mediante a presença de múltiplas restrições (TAVARES, 2002). Nesse sentido, projetos também podem ser definidos como sistemas abertos e complexos orientados a algum tipo de propósito (desenvolvimento de um produto ou processo, atender um pedido de um cliente, resposta a um choque tecnológico e ou mudança na legislação, entre outros). Ao absorver uma quantidade limitada de recursos, tais empreendimentos demandam o uso de técnicas gerenciais destinadas a integrar ações, em condições de risco e incerteza, objetivando garantir um equilíbrio entre as restrições primárias de escopo, prazo, custo e qualidade. Como apontado por Shenhar e Dvir (2010), as ações devem superar um conjunto mais amplo de restrições (novidade, complexidade, urgência e ritmo).

Cada uma das atividades de um projeto possui certo conjunto de atributos (relação de precedência, especificação de duração, definição de datas de início e término, estimativas de recursos, estimativa de custo, etc.). Em geral, uma vez especificado o escopo do produto e definidos os entregáveis do projeto, o processo de gerenciamento de projetos consiste em identificar as atividades e estabelecer a definição de atributos associados a cada uma dessas (PMI, 2008).

Diante da importância da definição dos demais atributos, uma tarefa crítica consiste na definição da estimativa e alocação de recursos para cada uma das atividades. Dada a disponibilidade de recursos e ou das alternativas técnicas (uso ou não de processos intensivos em tecnologia, por exemplo), tal ação sempre intervencionará a definição do prazo de duração bem como o custo de cada uma das atividades e do projeto como um todo (VASCONCELOS, 2007).

Em ambientes marcados pela necessidade de gerenciamento de vários projetos (doravante denominado ambiente multi-projeto – AMP) sujeitos a restrições de diferentes ordens (limitações de tempo, disponibilidade de recursos, restrições de ordem técnica, etc), o gerenciamento de portfólio de projetos – GPP deve fornecer critérios para a priorização dos

projetos (dados certos objetivos estratégicos) e, por conseguinte, deve orientar a alocação dos recursos sob o comando da organização. Assim, o GPP tem como objetivo viabilizar o gerenciamento dos diversos projetos de uma carteira em AMP, tendo em vista os objetivos estratégicos corporativos.

Um gerente de portfólio de projetos de uma organização que atua em AMP, ante a disponibilidade limitada de recursos (humanos, materiais, financeiros, etc) e restrições de ordem tecnológica, precisa levar em conta que, em certos momentos, diferentes projetos concorrem pelos mesmos recursos. Em outros esses recursos podem permanecer ociosos. Diante de tais fenômenos, a busca de economicidade exige a definição de critérios orientados à priorização de emprego dos recursos no GPP, fato que demanda grande esforço na definição e revisão dos atributos associados às atividades. Tal situação é mais crítica quando diferentes projetos de uma carteira se encontram em diferentes fases ou onde há grande fluxo de iniciação e finalização de tais empreendimentos. Nesse caso, como em um AMP cada projeto interage com os demais, tendo em vista a dotação limitada de fatores ou recursos, o gestor deve dispor de instrumentos de GPP que lhe permita dimensionar os resultados das decisões quanto à (re) alocação de recursos ordinários ou especialistas. Nesse caso, a depender da ação do gestor, a reprogramação de um único projeto pode causar um efeito em cascata nos atributos de duração, data de início e término, estimativa de custo, etc, não apenas em termos das atividades do projeto reprogramado, mas de todas as atividades de todo os projetos que fazem parte de uma carteira, e vice-versa. Abstraindo outros aspectos, um dos fatores que relaciona umbilicalmente diferentes projetos de um portfólio é a dotação limitada de recursos. Assim, sempre há um *trade-off* na busca de melhor desempenho quando um ou mais projetos é priorizado em detrimento dos demais (ARTTO, MARTINSUO e AALTO, 2001).

Recursos ordinários (máquinas ou recursos humanos não especializados) podem ser substituídos, (re) programados e ou (re) contratados com relativa facilidade. O grande desafio surge quando os diferentes projetos de um portfólio de projetos demandam ou concorrem por uma quantidade limitada de um mesmo recurso escasso ou que apresenta dificuldades em termos de substituição.

Do ponto de vista desse estudo, tal recurso será denominado como recurso especialista – RE. Um RE pode ser definido como: (i) um recurso escasso (que pode ser uma máquina ou recurso humano especializado), (ii) relativamente caro ou oneroso, (iii) de difícil contratação e ou substituição, (iv) que realiza tarefas específicas de alta complexidade necessárias à execução de atividades de um ou mais projetos, (v) que, sob o ponto de vista desses, apresenta acesso limitado (sendo disputado em diferentes atividades de diferentes

projetos) e que, por essa razão, (vi) demanda a fixação de critérios de acesso por parte do gerente de portfólio de projetos, os quais (vii) visam atender um dado objetivo do ponto de vista da GPP (minimização de custo, minimização de prazo, entre outros), que (viii) pode ou não ser mutuamente excludente (VASCONCELOS, 2007).

Com efeito, dada à definição de recurso especialista (RE) proposta anteriormente, o problema da Programação de Projetos com Restrição de Recursos (*Resource Constraint Project Scheduling Problem* – RCPSP) consiste na definição da Programação (*Scheduling*) das atividades de um projeto com o objetivo atingir um dado objetivo (VASCONCELOS, 2007). No entanto, tal como apontado por Laslo (2010), o RCPSP também se encontra relacionado ao GPP.

Não obstante, um dos principais problemas em RCPSP está associado ao *Job Shop Problem* – JSP. Tecnicamente o problema do JSP consiste em definir uma forma de garantir a alocação eficiente e eficaz de RE, sujeito a diferentes restrições (momento de iniciação do projeto, tipificação das relações de precedências, etc), fundamentalmente a presença de relações de precedências arbitrárias (*discretionary*).

Desse modo, JSP surge em função da interdependência entre diferentes projetos e está associado à necessidade de alocar uma quantidade limitada de recursos (fundamentalmente recursos especialistas ou RE) entre esses. Isso é particularmente importante em atividades econômicas orientados por projetos, tais como em certos segmentos ligados à indústria química, farmacêutica, mecânica, naval, exploração e produção (E&P) de petróleo e gás, produção de software, entre outras. Por exemplo, no caso de E&P, como definir quando e onde um valor esperado do risco – ERV irá operar dado que esse recurso especialista é demandado em diferentes locais de exploração.

Dadas as suas características o JSP é muito comum na indústria de software. Particularmente e segundo Pressman (2006) o desenvolvimento de softwares é caracterizado por três elementos: (i) produção sob encomenda, (ii) foco no desenvolvimento e (iii) ênfase no ciclo de vida do produto. A depender de sua complexidade, novidade, ritmo e ou urgência (SHENHAR; DVIR, 2010), o desenvolvimento de softwares pode exigir tanto o emprego de recursos humanos ordinários como também dada quantidade de recursos humanos especializados – RHE e caros (LASLO, 2010).

Muitas empresas especializadas na provisão de soluções tecnológicas na área de TI operam em ambientes que envolvem a execução simultânea de múltiplos projetos e demandam o uso de RE. Com efeito, precisam estar aptas para empregar e ou desenvolver metodologias destinadas ao planejamento e programação relacionados ao uso de tais recursos. Na verdade, o

grande desafio é aplicar critérios que garantam o equilíbrio entre os objetivos estratégicos e o melhor emprego de tais recursos, em condições de incerteza e risco.

Como apontado por Laslo (2010), sem a definição e implantação de métodos que orientem a tomada de decisões em GPP na presença de JSP, podem ocorrer disputas e a ingerência de *lobbies* internos que tendem a inviabilizar certas operações e aumentar os custos dos empreendimentos, uma vez que concorrem pelos mesmos recursos. Nesse contexto, as atividades de planejamento e o escalonamento de recursos especializada são apontados por Laslo (2010) como críticas no gerenciamento de AMP.

Não obstante, as atividades de gerenciamento de prazos, gerenciamento de custos, gerenciamento de riscos e sua integração são cruciais no processo de gerenciamento de projetos. Por exemplo, o gerenciamento de custo está diretamente relacionado ao gerenciamento de prazo por meio dos processos de estimativas de duração das atividades e sua estreita relação com as alternativas técnicas e estimativas de recursos prospectadas pelo gerente e ou equipe do projeto.

No entanto e em muitas situações, os resultados dos processos de gerenciamento estão diretamente relacionados ao grau de interdependência de um projeto com os demais projetos que conformam um determinado portfólio. Na verdade, como aponta Laslo (2010), o planejamento de recursos e a programação geralmente têm sido tratados de forma separada na literatura. Isso também se constata quando se aborda o tema de JSP em AMP, sob a perspectiva do gerenciamento de portfólio em projetos – GPP.

Diante do contexto descrito anteriormente, este trabalho se circunscreve à discussão de temas ligados a GPP em ambiente de desenvolvimento de software marcado pela presença de JSP com a utilização de recursos humanos especialistas em ambiente de múltiplos projetos.

1.1 FUNDAMENTAÇÃO DA PESQUISA

Orientadas ao alcance de eficiência (busca de custos reduzidos) e a eficácia (alcance de meta), as práticas de gestão de projetos – GP e de gerenciamento de portfólio de projetos – GPP se encontram cada vez mais associadas à: definição de alternativas e a melhor forma de alocação de recursos corporativos escassos entre fins alternativos (PMI, 2008); desenvolvimento mais rápido de produtos (PINTO; KHARBANDA, 1996); introdução de mudanças e inovações nas organizações (KERZNER, 2009; MARTINS, 2003); identificação de riscos e incertezas e gerenciamento de riscos (PMI, 2008).

Particularmente, as empresas que atuam no mercado de desenvolvimento de software, devido às características de tais atividades (produção sob encomenda, foco no desenvolvimento e ênfase no ciclo de vida do produto) devem possuir competências críticas. Como indicado na seção anterior, uma das mais importantes diz respeito à capacidade de planejamento, programação e de controle da execução dos projetos, fato que está diretamente relacionado à capacidade de alocação e emprego de recursos (internos e externos) de maneira exequível e confiável, evitando subutilização e ou estrangulamentos em função de competição interna pelo mesmo recurso ou mesmo sua má utilização. Devido ao alto conteúdo de urgência, complexidade e unicidade, possíveis ocorrências de desvios no orçamento e nos custos são recorrentes nessa indústria. Em parte esses estão associados ao que se denomina *job shop problem* – JSP, tal qual explorado nos estudos de Laslo (2010), Fattahi et. al. (2006), Ahmed et. al. (2004) Golenko-Ginsburg e Laslo (2004).

Fattahi et. al. (2006) definem o JSP como um problema em se determinar um cronograma de alocação de recursos com o sequenciamento de atividades pré-determinados em um ambiente de multi-projetos. Tal discussão sobre JSP também é apresentada por Ahmed et. al. (2004) como uma das mais complicadas tarefas em se tratando de alocação de recursos. Os autores definem o objetivo da alocação de recursos como a busca de um algoritmo para se efetuar a utilização de tais recursos de maneira eficiente. Dentro de um ambiente projetizado, o planejamento, alocação e o uso de vários recursos, inclusive recursos humanos especialistas, em um ou mais projetos se tornam complexos.

Ainda sobre alocação e utilização de recursos, Fattahi et. al. (2006) apontam a dificuldade de alcançar uma solução ótima para resolver este problema com o melhor desempenho de otimização possível, devido à alta complexidade do ambiente computacional. Por fim e de acordo com Fattahi et. al. (2006), as abordagens tradicionais de gerenciamento de projetos abordam esta questão de forma simples, tratando os recursos, inclusive os especialistas, sem restrição de alocações.

Mendes (2003) define que um dos principais benefícios em se utilizar de forma otimizada os recursos por um determinado período de tempo em um projeto é a liberação do recurso o mais rápido possível para alocação em outros projetos da carteira e redução dos riscos de não cumprimento dos prazos previamente estabelecidos.

Neste cenário, o grande desafio é definir a relação de precedência entre as atividades dos projetos e o tempo total de espera de alocação de um determinado recurso, pois existe, em

um ambiente de múltiplos projetos, a restrição de recursos e a necessidade de redução de prazo das atividades.

Finalmente, ao se tratar dos desafios associados à necessidade de balanceamento de recursos especialistas entre os múltiplos projetos. Laslo (2010) aponta que o grande problema é estabelecer uma regra que permita que a organização opere diferentes projetos em um ambiente de escassez dos recursos especialistas e demandas trazidas pela estratégia da empresa. Baker (1974) aponta problemas relacionados à alocação de recursos humanos especialistas ao longo do tempo pode ocorrer caso os processos de sequenciamento e escalonamento de atividades não forem efetuados de forma efetiva.

Feitas essas considerações, esse estudo procura avaliar as contribuições dos modelos propostos por Laslo (2010) – baseado em modelo de otimização, Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) – baseados em algoritmo genético, na busca da solução ótima do JSP em um ambiente de desenvolvimento de softwares. Os modelos propostos neste estudo procuram resolver os problemas de JSP referentes ao escalonamento de sequenciamento de atividades, tendo como agravante o problemas relacionados à alocação de recursos humanos especialistas.

Como resultado, este trabalho procura responder a seguinte questão:

“Quais as contribuições dos modelos de algoritmo genético e otimização para a resolução prática de problemas de Job Shop Problem em projetos de desenvolvimento de software com restrição de recursos humanos especialistas?”

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Analisar modelos de algoritmo genético e de otimização e suas contribuições para a resolução de *Job Shop Problem* em projetos de desenvolvimento de software com a utilização de recursos humanos especialistas.

1.2.2 OBJETIVOS ESPECÍFICOS

- a) Apresentar as vantagens e limitações dos modelos propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) para a solução do problema de *Job Shop Problem*;
- b) Realizar uma análise comparativa entre os modelos propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) para aplicações práticas de *Job Shop Problem*;
- c) Avaliar as propostas desenvolvidas em um estudo de caso de gerenciamento de projetos de desenvolvimento de software com restrição de RHE.

1.3 ESTRUTURA DO TRABALHO

A organização deste trabalho é composta por cinco capítulos: Introdução, Referencial Teórico, Metodologia, Resultado e Discussão, Conclusões e por fim, as Referências.

No Capítulo 1 foram apresentados os objetivos e as motivações deste estudo.

No Capítulo II – Referencial Teórico: neste capítulo serão abordados os principais conceitos e definições de gerenciamento de projetos, gerenciamento de portfólio de projetos, gerenciamento de riscos, projetos de desenvolvimento de software, características de JSP e modelos para resolução de JSP. Terá início pela contextualização do gerenciamento de projetos, seu relacionamento com o ambiente das empresas, seus processos em relação à alocação e sequenciamento de atividades e recursos. Depois irá abordar as técnicas e as características de gerenciamento de portfólio de projetos e do gerenciamento de riscos. Será apresentada a visão de gerenciamento de risco tendo em vista a relação com a alocação de recursos, custos e prazos.

Após essas definições, será caracterizado o ambiente de desenvolvimento de software, apontando os tipos de manutenções existentes na literatura, a apresentação do ciclo de vida e as características do gerenciamento de projetos nesta área. Na sequência são apresentados os conceitos e a classificação de Recursos humanos especialistas dentro do ambiente de desenvolvimento de software.

Finalmente será abordada a contextualização, modelos e a classificação de JSP. Para a resolução do JSP será apresentada os modelos matemáticos possíveis (modelos heurísticos e modelos baseados em algoritmo genético).

No Capítulo III – Metodologia: será trado o método de pesquisa a ser utilizado, enquadrado neste estudo como exploratório devido à integração entre as linhas de pesquisa relacionadas ao gerenciamento de prazo (planejamento e programação), gerenciamento de custo (em sua relação com o gerenciamento de prazos por meio do sequenciamento, estimativas de recurso, entre outras), gerenciamento de riscos (por meio de processo de estimação e simulação) e gerenciamento da integração. Serão apresentados os procedimentos metodológicos, com ênfase na criação e utilização de indicadores de aderência dos modelos propostos de resolução de JSP a um caso prático através de uma pesquisa semiestruturada ao gestor de uma empresa de desenvolvimento de software.

O Capítulo IV – Resultados e Discussão: é dedicado para a apresentação e discussão dos resultados do trabalho. Neste sentido, serão apresentados os modelos propostos e suas contribuições neste estudo para a resolução do JSP. São eles: modelo proposto por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004). Será aplicado o estudo em um caso prático em uma empresa de desenvolvimento de softwares.

A empresa será apresentada, contextualizada em relação aos problemas enfrentados pela área de desenvolvimento de software em relação aos problemas gerados pelo JSP. A seguir serão apresentadas as contribuições de cada modelo na percepção do gestor de projetos da empresa em estudo, bem como a conclusão em relação à aderência de cada modelo aos problemas enfrentados pelo gestor de projetos da empresa estudada. Finalmente, no capítulo V – Conclusões serão apresentadas os resultados das observações, as críticas e sugestões associadas bem como as contribuições, conclusões finais da dissertação, possíveis pesquisas complementares e a referência bibliográfica utilizada.

2 REFERENCIAL TEÓRICO

2.1 IMPORTÂNCIA DO GERENCIAMENTO DE PROJETOS E TIPOLOGIA DE AMBIENTES

Nas últimas décadas, mudanças econômicas, tecnológicas, político-institucionais sociais e culturais ocorrem de forma cada vez mais abrupta. Por essa razão, no último quartel do século XX um crescente número de organizações passou a reconhecer as vantagens e os desafios relacionados à adoção de práticas de gerenciamento em projetos – GP, gerenciamento de riscos em projetos – GRP e, mais recentemente, gerenciamento de portfólio de projetos – GPP. Tais práticas vêm crescentemente sendo introduzidas como parte essencial do arsenal de técnicas de gestão da mudança levadas a cabo pela necessidade da elaboração de projetos.¹

Segundo o PMI (2008), o GP está relacionado com a aplicação de conhecimentos, habilidades, ferramentas e técnicas em atividades não rotineiras destinadas a atender às necessidades e expectativas específicas, internas e externas à organização. No processo, a organização, o gerente e a equipe do projeto devem perseguir (i) o equilíbrio entre as demandas concorrentes de escopo, prazo, custo, qualidade e riscos, (ii) atendo-se à variáveis de influência ligadas ao ambiente interno e externo ao projeto.

A elaboração, análise e execução de projetos ocorrem em meio a um ambiente que pode ser classificado de diferentes formas. Sob o ponto de vista da organização, o ambiente externo (dinâmica de mercado e instituições que cristalizam as regras do jogo) e ambiente interno (estrutura organizacional, competências de base, disponibilidade, tecnologia empregada e tipos de recursos disponíveis, etc.) condicionam a estratégia, as metas e sua correspondente cristalização em atividades orientadas a novos empreendimentos (CAMPANÁRIO *et. al.*, 2009).

De maneira análoga, o PMI (2008) reitera a necessidade de considerar o ambiente cultural e social (influenciados pela educação, crenças e valores), internacional e político

¹As técnicas de gerenciamento de projetos têm sido estudadas e difundidas por duas associações bem conhecidas por profissionais que atuam nessa área. Uma delas é o *Project Management Institute* – PMI, a outra, o *International Project Management Association* – IPMA. Além disso, outras instituições também têm crescido em importância. Tal é o caso do *Institute for Operations Research and the Management Sciences* – INFORMS, *Decision Science Institute* – DCI e do *International Federation of Operational Research Societies* - IFORS.

(marcado por leis e normas) e ambiente físico (envolvendo aspectos ecológicos, geológicos, etc.) que podem impactar positiva ou negativamente um projeto. Nessa mesma linha de argumentação, Rabechini Jr. (2003) aponta a interferência do ambiente de *marketing* (calçado no conhecimento da dinâmica dos mercados), de conhecimentos técnicos e científicos (orientados à introdução de inovações) e de produção (condicionado paradigmas científicos e tecnológicos específicos que condicionam a forma como *inputs* são transformados em *outputs*).

Do ponto de vista deste estudo, dar-se-á grande importância ao ambiente interno. A rigor, o principal desafio dos gestores é garantir a alocação eficiente e eficaz de recursos em suas atividades, sejam rotineiras (operações) ou não rotineiras (projetos). Seja qual for a forma de classificação, o ambiente compreende um espaço amplo onde interagem diferentes fatores e ou forças econômicas e sociais que condicionam as escolhas, o emprego de recursos e o êxito dos empreendimentos. Tais forças interferem na escolha, viabilidade, formatação do(s) projeto(s) e em sua(s) provável(s) chance de sucesso ou fracasso (CLELAND; IRELAND, 2004), interferindo direta e indiretamente na priorização de ações e emprego de recursos organizacionais entre fins alternativos com implicações sobre certos parâmetros do projeto, tais como duração, níveis de custo, riscos, etc.

Um aspecto crítico, diretamente associado ao ambiente interno, é que diferentes projetos concorrem por uma quantidade limitada de recursos humanos e materiais (CAMPANARIO *et. al.*, 2009). Seja como for, a análise, a identificação e a avaliação das variáveis do ambiente interno são críticas uma vez que sua interação pode afetar o desempenho dos projetos. Por essa razão devem compreender temas recorrentes e centrais no âmbito da teoria e prática em GP e GPP. Também o papel da informação e seu uso nos processos de planejamento são críticos. A rigor, um projeto pode ser definido como um conjunto de informações (internas e externas à empresa) coletadas, processadas e analisadas para a tomada de decisões em condições específicas de risco e incerteza. (WOILER; MATHIAS, 1996)

2.2 GERENCIAMENTO DE PROJETOS

Práticas de GP fornecem subsídios (conceitos, métodos, modelos, técnicas e ferramentas) orientados ao planejamento e emprego de recursos organizacionais no contexto de risco e incerteza como meio de assegurar o alcance de melhores resultados em atividades

não rotineiras. Tal fato é apontado por diversos autores entre eles Gilb (1990), Pinto e Kharbanda (1996), Clarke (1999), Sandeep (2002), Vieira (2002), Andersen e Jessen (2002), Martins (2003), Neto e Bocoli (2003), Rabechini Jr., Carvalho e Laurindo (1999), Campanário *et. al.* (2009) e Kerzner (2009).

Para Clarke (1999), por exemplo, em ambientes dinâmicos ferramentas de gerenciamento de projetos (GP), se usadas de forma sistemática, podem garantir a eficiência (alcance de meta) e a eficácia (atender a restrições de custos) na gestão de recursos escassos entre fins alternativos. Já Vieira (2002) assevera que o ambiente de atuação das empresas demanda novas práticas por parte das organizações e que é comum que essas estejam relacionadas à introdução e aos resultados de projetos. Kerzner (2009) também aborda o assunto. Ao fazer isso, aponta que gerenciar de maneira eficiente em um ambiente de grandes mudanças constitui o grande desafio a ser vencido pelos gestores contemporâneos. Assim, como meio de dar conta dos desafios associados à dinâmica de mercado, social e técnica em curso, é preciso estar preparado para efetuar o GP de forma tecnicamente consistente e sustentável do ponto de vista econômico e financeiro, fato que envolve o emprego de abordagens para dar conta das incertezas e riscos inerentes às operações rotineiras e não rotineiras.

Neto e Bocoli (2003), Martins (2003) e Sandeep (2002) destacam que o GP compreende atividade relativamente nova e emergente, motivada pelo fato de as organizações (públicas e privadas, instituições de ensino, etc.) estarem buscando cada vez mais estudar, difundir, compreender e desenvolver métodos, práticas e ferramentas nesta área. Para Kerzner (2009) os princípios e as práticas GP são cada vez mais utilizados para operacionalizar a implantação dos planos estratégicos bem como otimizar o fluxo de trabalho horizontal e vertical nas organizações e correspondente alocação de recursos. Assim, a adoção de práticas de GP constitui meios para a implantação de todo o tipo de plano. Desde a formulação do planejamento estratégico (PE) até a sua efetivação, por meio de portfólio e projetos, técnicas de GP auxiliam na implementação de decisões no fluxo organizacional, alocação de recursos escassos e sua integração (na organização e em diferentes projetos) em contextos distintos (CAMPANÁRIO *et. al.*, 2009).

Enquanto esforços temporários orientados à implantação de novos produtos no mercado e ou processos nas organizações, o GP é crítico na gestão de mudanças e na introdução de inovações (de produto ou processo) nas organizações e no tecido econômico (TAVARES, 2002). Ainda segundo Gilb (1990), os processos de gestão se baseiam em planos (de projeto e de período). Tais planos, circunscritos por um ambiente cada vez mais dinâmico

e incerto, são apreendidos por estimativas, aproximações, “tratamento” de incertezas e, conseqüentemente, modelagem de riscos. A conceituação e identificação dos riscos existentes para realização de um projeto, bem como o estudo de técnicas, emprego de critérios de avaliação e de meios destinados a evitá-los e ou amenizá-los, são tarefas essenciais para o gerenciamento de qualquer de plano ou projeto e, por conseguinte, estão associadas à taxa de êxito de tais empreendimentos, bem como sobrevivência e crescimento de uma organização. Tais técnicas também devem ser aplicadas no GPP como meio de garantir a consecução dos objetivos estratégicos, redução da incerteza e melhora na relação risco/retorno corporativo. Dessa forma, a existência do risco é um fator inerente a qualquer empreendimento devendo ser abordada por métodos e técnicas específicas como meio de maximizar as chances de sucesso e ou minimizar a possibilidade de fracasso dos empreendimentos.

Em suma, do ponto de vista operacional, o GP compreende esforços destinados à criação de produto(s) e ou serviço(s) único(s) por meio de esforços temporários em meio a um contexto mais amplo que o do próprio projeto (KERZNER, 2009). Assim, as práticas de GP cada vez mais se encontram associadas à identificação e o melhor uso de recursos corporativos escassos, desenvolvimento mais rápido de produtos, (PINTO; KHARBANDA, 1996) bem como com a introdução de mudanças e inovações (MARTINS, 2003), no contexto de riscos e incertezas internos e externos á organização (PMI, 2008).

Como sugerido vale salientar que a importância do gerenciamento dos riscos extrapola a necessidade de aplicação de processos customizados destinados a garantir o sucesso de um único projeto. Em ambientes competitivos, marcados por crescente ritmo de introdução de novos produtos e processos, as organizações devem gerenciar de forma eficiente, eficaz e simultânea um número cada vez maior de projetos (CLARK; WHEELWRIGHT, 1993; COOPER, 1993). Em meio a rápidas mudanças a tendência é que se passe a dar crescente importância aos processos de GPP o que inclui o uso de novas técnicas de (re) avaliação e de priorização de projetos (KERZNER, 2009).

2.2.1 GERENCIAMENTO DE PORTFÓLIO DE PROJETOS

Pressionadas pela urgência, muitas organizações avaliam, aprovam e priorizam seus projetos de forma *ad hoc* e individualizada, dando maior ênfase na determinação de custo e dimensionamento de prazos. Pouca ênfase é dada à necessidade de planejamento e ou programação integrada dos empreendimentos. Dai a importância da introdução de processos

de GPP. Na verdade, em ambientes marcados pela presença de riscos e incertezas, todas as organizações, mesmo aquelas que não adotam processos formais, tendem a praticar algum tipo de gestão de portfólio (COOPER, 1993). Este visa garantir que o gerenciamento seja feito de forma mais integrada visando minimizar riscos e potencializar oportunidades; sobretudo, aquelas atreladas às competências de base e aos objetivos estratégicos de curto, médio e longo prazo.

Com efeito, não é exagero afirmar que o ápice da gestão de portfólio consiste em estabelecer critérios destinados a maximizar/aperfeiçoar o aproveitamento dos recursos e competências disponíveis primando pela busca de equilíbrio entre a alocação e o uso de recursos escassos entre fins alternativos, a otimização de resultados (atendimento do escopo com custos e prazos reduzidos em um ambiente multi-projetos), bem como o monitoramento e o controle dando conta de incertezas e riscos, em um ambiente marcado pela existência de múltiplos projetos (COOPER, 1993).

Assim, a gestão de portfólio de projetos fornece um arsenal de técnicas e ferramentas destinadas à seleção e priorização de projetos e essas vêm se tornando fundamentais para a consolidação de dados e informações empregadas na tomada de decisões, incluindo a alocação de recursos especializados escassos em atividade não rotineiras. A disciplina fornece recursos (conceitos, modelos/‘algoritmos’, indicadores, representações visuais, etc) orientados à caracterização, comparação, avaliação, orientadas à seleção e ou priorização dos projetos (KERZNER, 2009).

O portfólio de projetos compreende “... uma coleção de projetos ...” que podem “... estar relacionados ou serem independentes dos demais ...” e que, em muitas situações, “..compartilham dos mesmos objetivos estratégicos e competem pela utilização dos mesmos recursos” (ARCHER; GHASEMZADEH, 1998, p. 1). Esse último aspecto é particularmente importante em segmentos econômicos orientados por projetos, tais como aqueles ligados às indústrias químicas, farmacêutica, mecânica, naval, exploração e produção (E&P) de petróleo e gás, produção de software, entre outras.

Do ponto de vista operacional, os modelos de gestão de portfólio permitem a utilização de diferentes técnicas cuja escolha deve levar em conta a facilidade de uso, a escassez de tempo, a disponibilidade de recursos internos, a familiaridade da equipe com os modelos e os resultados que se espera obter (ARTTO, 2001). Definir e gerenciar um portfólio de projetos exige o emprego de modelos destinados à priorização dos diversos empreendimentos. Isso exige o emprego de técnicas que resultem em critérios destinados orientar os processos de reprogramação e alocação de recursos humanos e materiais. Sendo

assim, gerenciar uma carteira de projetos exige a aplicação de métodos que garantam ao administrador vencer certos desafios. Em geral, o gerenciamento de uma carteira de projetos se dá em um ambiente marcado pela insuficiência de recursos em relação às necessidades e ao número de projetos, escassez de informações e excesso de projetos secundários (COOPER *et al.*, 2000). Esses aspectos demandam o emprego de critérios de priorização e implementação de um enfoque de administração sistêmico, focado na constante (re) avaliação da disponibilidade de recursos (*inputs*) e nos resultados esperados (*outputs*). No entanto, a elaboração e análise de cronogramas e orçamentos em AMP na presença de RE ainda são exploradas de forma incipiente na literatura de GP e GPP.

Feitas essas considerações sobre os desafios do ambiente, importância do GP e do GPP, a próxima seção explora aspectos relacionados existência de incerteza e riscos em projetos com ênfase no problema da alocação de recursos.

2.2.2 GERENCIAMENTO DE RISCOS

Em meio às mudanças no paradigma técnico-científico que marcou a década de 70, a relação entre incerteza e risco foi explorada por Meneses (2001). Ao considerar implicitamente a relação risco/retorno e a importância de práticas de gestão orientadas aos riscos, esse autor salienta que “... *uma melhor performance econômica só é alcançada por meio de um aumento no grau de incerteza, ou seja, elevando-se os riscos ...*” (MENESES, 2001, p. 29). Como é impossível eliminar todos os riscos, do ponto de vista organizacional “... *é essencial que os riscos selecionados sejam os riscos certos...*” (MENESES, 2001, p. 29). Assim, o gestor deve estar disposto “... *a escolher racionalmente os riscos e as ações associadas, ao invés de identificar as incertezas baseados na intuição, não importa o quanto os riscos serão quantificados*” (MENESES, 2001, p. 29).

Embora o tema explorado por Meneses (2001) não seja novo ele é recorrente. Como sugerido acima, o tratamento dos riscos sempre deve estar no cerne das decisões dos gestores. Especificamente, no campo do GP o emprego de boas práticas deve assegurar o equilíbrio entre os requisitos de escopo, tempo e custo, os quais compreendem o triângulo restrições primárias de qualquer empreendimento (RABECHINI JR.; CARVALHO; LAURINDO, 2009). Na verdade, esses aspectos sempre estão de alguma forma associados a outros temas em GP como qualidade, integração, comunicação, etc., em meio à presença de incertezas e riscos (PMI, 2008).

Com efeito, o sucesso na gestão de um projeto está relacionado à definição e ao alcance dos objetivos pré-estabelecidos. Esses dizem respeito à entrega dos resultados do projeto dentro do prazo previsto, custo orçado, com nível de desempenho esperado e aceitação pelo cliente.

O alcance de tais objetivos exige a implantação de ações destinadas a garantir o controle de mudanças de escopo, custo, prazo, etc, respeitando a dotação de fatores e cultura da organização. Grande parte das ações se baseia no planejamento orientado à identificação, avaliação e tratamento de fatores de risco associados aos empreendimentos (PMI, 2008).

Nessa linha de investigação, Koontz e O'Donnell (1980) definem que gerenciar consiste em executar tarefas/atividades para atingir determinado propósito por meio de sistemático planejamento e controle para que objetivos previamente definidos sejam alcançados com o mínimo de desvios. Assim, todas as etapas e ações no âmbito de um projeto, desde sua iniciação até a fase de finalização, remetem de uma forma ou de outra ao gerenciamento de riscos.

A disciplina gerenciamento de riscos em projetos tem crescido em importância, mas ainda é incipiente (KWAK; STODDARD, 2004). Por essa razão tem sido objeto de investigação por diversos autores, entre eles Chapman e Ward, (1997), Wideman (1992), Meyer, Loch e Pich (2002), Lyons e Skitmore (2004), e Shenhar e Dvir (2010). Uma importante contribuição na discussão do tema é dada por Perminova, Gustafsson e Wikström (2008) que destacam a necessidade de melhor definição do conceito de incerteza no ambiente de GP.

Segundo Silva, Martins e Rabechini Jr. (2011), ao propor modelo destinado à integração de avaliação de risco de custo e de prazos em projetos, uma das questões que merece atenção nos estudos sobre gerenciamento de risco em projetos refere-se ao estudo aos processos de gerenciamento de riscos e sua integração com outras áreas ou disciplinas em gerenciamento de projetos, sobretudo no que tange a integração das áreas de gerenciamento de custo e gerenciamento de prazo. Nesse contexto, o gerenciamento de risco constitui a prática de usar técnicas de análise e de avaliação de riscos como meio de desenvolver estratégias de gestão para que esses sejam reduzidos e ou amenizados (GALWAY, 2004).

Tendo em vista esses aspectos, Galway (2004) aponta que antes de um projeto iniciar e enquanto este estiver em progresso, respostas a perguntas como "quanto tempo o projeto irá demandar?", "qual será seu custo final?" e "o produto atenderá às especificações previstas?" são incertas, fato que remete à necessidade de modelagem de incertezas e riscos. A rigor, respostas parciais a essas questões são delineadas na fase de planejamento do projeto, o que

inclui obtenção de estimativas de duração das atividades (que leva em conta as estimativas de recursos) bem como as estimativas de custo das atividades (que, dentre outras entradas, envolve o uso das estimativas de duração), necessárias à elaboração do orçamento. Além disso, a implementação do plano do projeto demanda o monitoramento e o controle bem como a verificação e atualização sistemática desse. Por essa razão, processos de iniciação, planejamento, execução, monitoramento e controle e finalização, se repetem sistematicamente ao longo do ciclo de vida do projeto. Da mesma forma, técnicas de gerenciamento de riscos devem ser aplicadas durante todo o ciclo de vida do projeto como meio de dar conta dos riscos e das incertezas inerentes ao projeto, bem como aspectos não previstos no plano original, dentre esses as necessidades de priorização de projetos e de realocação de recursos entre esses.

De forma análoga, Daneshkhah (2004) levanta as seguintes questões: “como os riscos podem ser efetivamente identificados?”; “como esses podem ser geridos?”; “sob quais circunstâncias esses devem ser aceitos ou rejeitados?”; “como diferentes *stakeholders* interpretam e ou reagem ante aos eventos de riscos?” As respostas a essas questões se relacionam aos processos de identificação, avaliação, disponibilidade e custo de informações internas e externas a empresa, estando ainda atreladas às competências (conhecimentos, habilidades e atitudes) do gerente, da equipe de projetos e dos demais *stakeholders* (RABECHINI JR.; CARVALHO; LAURINDO, 2009) e também a aspectos comportamentais e cognitivos que orientam gestores e tomadores de decisão em suas atividades (VIRINE; TRUMPER, 2008). Especificamente, “avaliação de risco é amplamente reconhecida como um processo sistemático quantitativo” (DANESHKHAH, 2004 p. 1). Dessa forma, “risco é comumente descrito como uma combinação da probabilidade de um evento indesejável (acidente) ocorrido e suas consequências.” (DANESHKHAH, 2004 p. 1).

Finalmente, Kaplan (1997) expressa o risco como uma combinação entre probabilidade de ocorrência de um dado evento e a avaliação de sua respectiva consequência ou severidade. Já a incerteza representa ignorância ou falta conhecimento acerca da ocorrência, distribuição de probabilidade e respectivo impacto associado a certos eventos. Expressando de forma sumária as definições: incerteza pode ser definida como a parcela do risco que não pode ser medida; não obstante risco pode ser conceituado como a parcela mensurável da incerteza.

Tradicionalmente, o termo risco é empregado quando nos referimos a aspectos mensuráveis da incerteza – sendo aquele definido como a parcela da incerteza que não podemos mensurar. Assim, ao procurarmos definir risco, tautológica estaremos definindo

incerteza (SILVA; MARTINS; RABECHINI JR, 2011). Tal dificuldade nos remete à necessidade de uma maior reflexão acerca de tais temas no contexto de GP (PERMINOVA; GUSTAFSSON; WIKSTRÖM, 2008).

Do ponto de vista desse estudo, AMP apresentam forte interdependência em função da disputa por RE e, em muitas situações, esses aspectos são negligenciados. Tal fato tende a afetar o grau de exposição a riscos e incertezas.

2.3 CARACTERÍSTICAS DE PROJETOS DE DESENVOLVIMENTO DE SOFTWARES

Ante a emergência da adoção de GP e GPP nas organizações e necessidade da adoção de métodos sistemáticos como meio de dar conta da incerteza e dos riscos intrínsecos a qualquer empreendimento, o grande desafio do gestor está associado seleção e implantação de métodos de gestão e de modelagem dos riscos em atividades não rotineiras. Já a modelagem de riscos compreende o uso de técnicas destinadas a identificar e avaliar a ocorrência de eventos que podem afetar positiva ou negativamente o desenvolvimento do projeto de forma sistemática a fim de atribuir-lhes medidas de probabilidade e de severidade. Em suma, o gerenciamento de riscos em GP e GPP devem combinar técnicas de levantamento e tratamento de sistemático de informações como meio dar subsídios ao gerenciamento riscos. Os processos envolvidos visam dar conta de uma série de questões, tais como as apontadas por Galway (2004) e Daneshkhah (2004).

Em certas indústrias, devido às suas particularidades inerentes, há um esforço no sentido de desenvolvimento de abordagens customizadas em GP – embora, como será indicado, poucos esforços sejam empreendidos em GPP. Tal é o caso da indústria de softwares, que foi objeto de estudos por parte de diversos autores, entre eles: Boehm (1988), Bhoem (1990), Boehm e DeMarco (1997), Fairley (1994) e Presman (2006).

Bhoem (1990), Boehm e DeMarco (1997) destacam a dificuldade em se definir o escopo do software na fase de concepção e elaboração do projeto, através de técnicas de levantamento de requisitos junto ao cliente. De acordo com Boehm (1988), a fase inicial de documentação das necessidades do cliente não deve ser um processo “desgastante” para se atingir seus objetivos, além de obter melhorias na elaboração do escopo. Já Fairley (1994) argumenta que muitos projetos de software não são entregues dentro do prazo acordado e dentro do custo planejado por não possuírem um modelo eficaz de gestão de risco.

Com a utilização de softwares cada vez mais complexos devido ao baixo custo de hardware, Pressman (2006) aponta que um dos maiores problemas enfrentados pelos profissionais desta área é a dificuldade de se gerenciar, ao invés dos problemas técnicos.

Aspectos centrais em negócios ligados a indústria de software são apontados por Pressman (2006). Segundo esse autor, projetos associados a essa indústria se originam a partir das seguintes circunstâncias: *necessidade de manutenção corretiva* (processo que inclui o diagnóstico e a correção de erros do programa após o produto de software já ter sido entregue – erros remanescentes), *necessidade de manutenção adaptativa* (adapta/modifica o produto de software para que ele tenha uma interface adequada de acordo com as diversas mudanças de ambiente – hardware e/ou seus softwares operacionais), *necessidade de manutenção evolutiva* (atividade de modificar o sistema para atender novas capacidades e funcionalidades, para que o produto de software não se torne obsoleto) e *necessidade de manutenção preventiva* (o produto de software é modificado para melhorar suas características de confiabilidade ou de manutenções futuras). Em geral, tais manutenções derivam de necessidades específicas de um negócio (PRESSMAN, 2006).

O Quadro 1 apresenta os tipos de manutenções em produtos de software e suas principais características. A manutenção corretiva não possui uma periodicidade definida. Caso ela ocorra, o grau de severidade do risco envolvido deve ser analisado e, em caso de necessidade, será preciso a alocação de um recurso especialista para execução desta manutenção. Já as manutenções preventiva, adaptativa e evolutiva devem ser planejadas e devem possuir recursos especializados, visando manter o software com o maior índice de disponibilidade possível. Devido à natureza do produto, que visa atender necessidades específicas em diferentes contextos, a indústria de software se caracteriza por elevado grau de projetização. De fato, certas características comuns do processo de desenvolvimento de softwares contrastam com o processo de desenvolvimento de produtos manufaturados. Segundo Pressman (2006) o desenvolvimento de softwares é caracterizado por três elementos: (i) *produção sob encomenda*, (ii) *foco no desenvolvimento* e (iii) *ênfase no ciclo de vida do produto*. Essas características e suas implicações são discutidas sumariamente a seguir.

Quadro 1 - Tipos de manutenções de software.

Tipo de manutenção	Característica
Corretiva	Correção de erros remanescentes – cuja finalidade é a correção de problemas na versão do software atual do cliente.
Adaptativa	Adaptação do produto às possíveis mudanças de hardware e ou software – cuja finalidade é efetuar alterações no software do cliente

	em relação a possíveis problemas de desempenho ou incompatibilidade de hardware e/ou software.
Evolutiva	Alteração de requisitos demandados pelo cliente – cuja finalidade é a adequação do software em relação a alterações ou inclusões de requisitos demandados pelo cliente.
Preventiva	Modificação de certos requisitos do sistema – cuja finalidade é a programação de alterações preventivas à possíveis falhas no software.

Fonte: Elaborado pelo autor com base em Pressman (2006).

Empresas que operam em indústrias que produzem sob encomenda devem possuir certas competências críticas: boa capacidade de planejamento e de controle da produção; criação ou adoção de ferramentas que sustentem a programação das ordens de produção; garantia de alocação e emprego de recursos (internos e externos) de maneira exequível e confiável, evitando subutilização e estrangulamento nos processos de execução (causados por competição interna pelo mesmo recurso); desenvolvimento de competências específicas em engenharia de projeto, produto e serviços; capacidade comercial e de oferta de serviços de assistência técnica pós-venda, entre outras (PRESSMAN, 2006).

Na verdade, as competências relacionadas acima constituem elementos vitais para que as empresas que operam em regime de encomenda consigam atender de maneira satisfatória critérios associados a prazos, custos e de qualidade. Embora os desenvolvedores de software possuam uma postura cada vez mais inclinada à montagem baseada em componentes, a maior parte dos softwares continua sendo desenvolvida sob encomenda, uma vez que um software se destina ao atendimento de necessidades específicas exigindo o desenvolvimento de novos sistemas, subsistemas e componentes (PRESSMAN, 2006).

A Figura 1 apresenta o modelo clássico do ciclo de vida de desenvolvimento de softwares, tal como sugerido por Pressman (2006). A fase de planejamento global objetiva definir, de forma preliminar, o escopo do software, bem como suas restrições e conceitos (levantamento de requisitos junto ao cliente visando compor as regras de funcionalidade a serem atribuídas ao software). Na fase de análise, visa estabelecer critérios e elaborar as documentações lógicas que serão utilizadas nos processos de confecção do software, bem como os protótipos e relatórios necessários. Durante a fase de planejamento, devem-se definir os elementos de hardware e software necessários para a execução do projeto. Nesta fase também devem ser definidas as tarefas que cada recurso irá executar e o refinamento dos documentos efetuados na fase anterior (destaque-se que do ponto de vista do sucesso do projeto tal fase é crítica).

Na fase de codificação, os códigos devem ser produzidos respeitando a documentação previamente produzida. Na fase de testes, todos os componentes do software devem ser rigorosamente testados, em relação ao que foi planejado, em busca de erros de codificação e de interpretação, inclusive testes de desempenho. Por fim, na fase de manutenção, devem ser executadas as manutenções decorrentes de erros, mudanças oriundas das necessidades do cliente, legislações, entre outros.

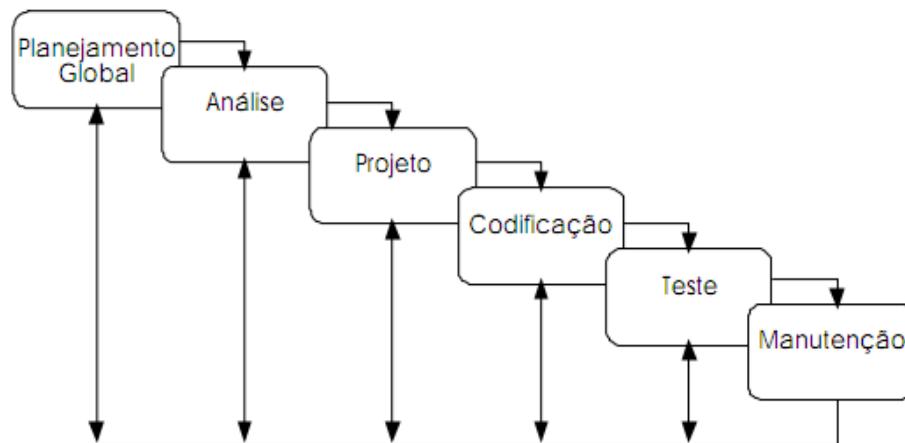


Figura 1 - Ciclo de vida clássico do desenvolvimento de software.
Fonte: Adaptado de Pressman (2006, p. 33).

Assim, segundo Pressman (2006), o processo de desenvolvimento do software exige a confecção de documento que servem de apoio ao levantamento e definição de requisitos e funcionalidades específicas do cliente ao qual devem ser implantadas ao novo software sendo desenvolvido. Com o escopo de todas as funcionalidades do *software* definido (fases de planejamento e análise), o centro da tomada de decisões se volta aos processos relacionados na fase de projetos. Em contraste com manufatura, cujo uso e o acesso a componentes padronizados (por exemplo, produtos que possuem dada capacidade de armazenamento de dados e ou de memória passíveis de emprego em diversas linhas de produto), os quais permitem que engenheiros e projetistas se concentrem nos elementos realmente inovadores do projeto, projetos de desenvolvimento de software são elaborados de forma mais integrada, em ciclos e fases mutuamente dependentes e sujeitas a verificações. Esses compreendem a integração de partes, peças e componentes destinados à construção de subsistemas e sistemas que irão compor o produto final – software definitivo.

A importância da integração de sistemas é crítica – ponto que é explorado por Chagas e Cabral (2010) ao abordar a indústria aeroespacial e por Chagas Junior e Duarte (2011). Tal plasticidade tende a gerar ganhos. No entanto esses não estão relacionados a economias de escopo e de escala, mas se baseiam em economias de repetição de recombinação. Mesmo que não leve em conta tal insight, segundo Presman (2006) no mercado de desenvolvimento de softwares o emprego a integração de partes, peças e componentes padronizados ainda se encontram em fase embrionária (PRESMAN, 2006). A depender do projeto, a customização necessária ao atendimento de certos requisitos demanda grande esforço no entendimento das funcionalidades e desenvolvimento customizado de subsistemas e sistemas, sendo os ganhos ou economias de escala são ainda insipientes.

Em suma, a produção sob encomenda, característica na indústria de software, implica unicidade no processo de desenvolvimento, grande esforço no levantamento de requisitos/funcionalidades do produto exigindo *expertise*, alto grau de capacitação técnica e competências em gerenciamento de projetos. Devido ao alto conteúdo de urgência, complexidade e unicidade, riscos de desvios no orçamento e nos custos são recorrentes nessa indústria.

Como será explicitado neste estudo, estes fatores estão associados ao que se denomina *job shop problem – JSP*, tal qual explorado nos estudos de Laslo (2010), Fattahi *et. al.* (2006), Ahmed *et. al.* (2004) Golenko-Ginsburg e Lasto (2004), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004).

Adicionalmente, softwares não são manufaturados no sentido clássico – fato que corrobora em certa medida a visão de Chagas Junior e Duarte (2011). Por essa razão, os processos de desenvolvimento de softwares não podem ser geridos por meio de processos tradicionais, empregados na manufatura. Os esforços no desenvolvimento de softwares se orientam mais na direção do *design* de subsistemas e sistemas que têm suas definições apresentadas no Quadro 2.

A organização dos projetos de desenvolvimento de software contrasta com projetos destinados ao desenvolvimento e produção de bens seriados cujo foco do processo se encontra mais orientado a requisitos tangíveis tais como: miniaturização, velocidade de processamento, capacidade de memória, entre outros, como no caso de desenvolvimento de hardwares. Com efeito, o foco no desenvolvimento, particularmente no design, é característica marcante em projetos ligados à indústria de softwares.

Quadro 2 - Características de sistema e subsistema.

Sistema	<p>"Sistema é um conjunto de partes coordenadas para realizar um conjunto de finalidades." (CHURCHMAN, 1972, p.50);</p> <p>"Conjunto de partes e componentes, logicamente estruturados, com a finalidade de atender a um dado objetivo." (CASSARRO, 1988, p. 27);</p> <p>"Uma rede de componentes interdependentes que trabalham em conjunto para tentar realizar um objetivo." (DEMING, 1997, p.41).</p>
Subsistema	<p>Os sistemas podem ser divididos em partes menores denominadas subsistemas e, estes, em rotinas, sub-rotinas e aí por diante, buscando atingir determinado objetivo. A questão é abordada por Cautela e Polloni (1982) quando afirmam que qualquer sistema pode ser encarado como subsistema de um sistema maior, sendo isto denominado hierarquia de sistemas.</p> <p>Nakagawa (1993, p.23) afirma: "devido ao fato de que os subsistemas, em graus variáveis, são interdependentes, as modificações ocorridas em um subsistema, provavelmente, afetam o comportamento dos outros subsistemas".</p>

Fonte: Elaborado pelo autor com base em Churchman (1972), Cassarro (1988), Deming (1997), Cautela e Polloni (1982) e Nakagawa (1993).

De acordo com Pressman (2006), projetos de softwares possuem ciclo de vida curto e, ao longo de seu ciclo de vida/uso, podendo sofrer novas exigências de manutenção como meio de atender à legislação corrente. Também esses podem demandar melhorias e possíveis implementações para eventuais correções de erros. Softwares bem elaborados passam por estágios de concepção (onde se responde a pergunta “o que:=?”), de desenvolvimento (“como?”) e por último as mudanças (de ambiente) no sistema.

Finalmente, softwares não “se desgastam” no sentido em que esses produtos não são suscetíveis à depreciação no sentido tradicional do termo. Dado um produto existente, aprimoramentos e desenvolvimentos de novas funcionalidades sempre são passíveis de obtenção. Por essa razão, Tao (2008) aponta que na indústria de software muitos projetos visam aprimorar as funcionalidades de um produto existente.

Em suma, projetos de desenvolvimento devem gerar, de forma econômica (baixo custo/alocação eficiente de recursos), rápida (respeitando as restrições de recursos e de prazo) e confiável (atendendo a especificações e requisitos), um produto novo ou ‘aperfeiçoado’ por meio de avanços incrementais ou mesmo por meio do desenvolvimento de novas funcionalidades associadas a um produto existente. Por essas razões, há grande ênfase no ciclo de vida do produto sendo que a capacitação em projetos é a grande responsável pelo

desenvolvimento contínuo, viabilidade de soluções customizadas e sucesso em tais empreendimentos (TAO, 2008).

Um dos modelos mais conhecidos e fluentes em se tratando de desenvolvimento de software é o modelo em espiral, proposto por Boehm (1998), demonstrado na Figura 2. Este modelo foi desenvolvido para abranger as melhores características do modelo de ciclo de vida clássico, bem como estabelecer um meio mais contumaz de análise de riscos, por meio de contínua verificação de resultados ao longo do ciclo de vida do projeto. Esse aspecto é central na medida em que nas fases de desenvolvimento de softwares erros tendem a se acumular em escala exponencial demandando custosos processos de correção. No modelo de Boehm (1998), em cada estágio do ciclo em espiral são incluídos processos de comunicação/validação com o solicitante (cliente). As saídas desses processos demandam novos ciclos de planejamento, análise e projeto, prototipação e avaliação. A trava de monitoramento e controle de riscos está baseada nos processos cíclicos de verificação e validação em cada estágio. Todos os processos devem ser sistematicamente repetidos até que o software seja entregue ao cliente.

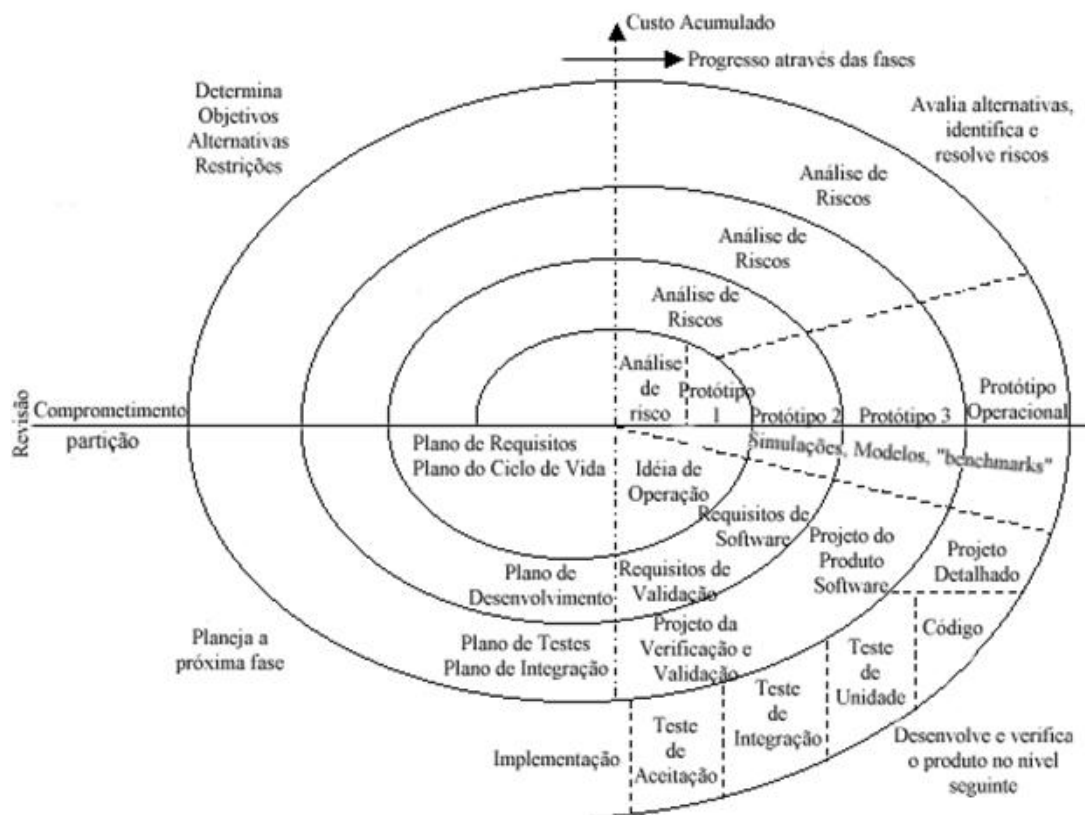


Figura 2 – Modelo em espiral de desenvolvimento de software.
Fonte: Boehm (1998, p. 3).

Em função das três características discutidas anteriormente (produção sob encomenda, foco no desenvolvimento e ênfase no ciclo de vida do produto), o desenvolvimento de softwares apresenta elevado grau de exposição a incertezas e riscos. Na literatura, esses são abordados e tipificados de diferentes formas. Em que pese à existência dos modelos customizados ao gerenciamento de projetos de softwares, cujo expoente mais representativo é Boehm (1998), parte da literatura relacionada ao tema enfatiza a tipificação de riscos e suas implicações no gerenciamento de projetos de desenvolvimento de softwares. Ao fazer isso, aproximam-se mais de abordagens tradicionais, sem se ater as especificidades relacionadas à produção sob encomenda ligada aos processos de desenvolvimento de softwares.

De forma geral, o gerenciamento de risco do projeto de software pode ser definido como emprego sistemático de técnicas de gestão, de procedimentos e de práticas como meio de identificar, analisar, planejar e gerenciar riscos como meio de minimizar as ameaças e maximizar as oportunidades (XIAOSONG *et al.*, 2009). Tal definição vai ao encontro direto a abordagem tradicional de gerenciamento de riscos preconizada pelo PMI (2008). Segundo Xiaosong *et. al.* (2009), o processo de gerenciamento de riscos pode ser feito por meio de 4 (quatro) processos: identificação (pesquisa, listagem e caracterização dos elementos de risco); análise (priorização dos riscos identificados através de critérios estabelecidos); planejamento (selecionar e implementar as opções de tratamento de risco); gerenciamento (implantação, monitoramento por relatórios, revisão ações de gestão dos riscos em relação aos objetivos do projeto). A relação entre esses processos é ilustrada na Figura 3.

De acordo com Galway (2004), a análise de risco é o processo de avaliação dos riscos, enquanto o gerenciamento de risco utiliza as saídas do processo de análise de risco como meio de estabelecer e ou desenvolver estratégias de gestão orientadas para que os riscos sejam reduzidos e ou amenizados. Em gerenciamento de projetos, muitas das técnicas são utilizadas para abordar as seguintes questões: “qual a duração do projeto?”, “qual o custo do projeto?”; “produto atenderá as especificações?”. A abordagem dessas remete desenvolvimento e ao tratamento dos riscos de prazo, riscos de custo e riscos de desempenho.

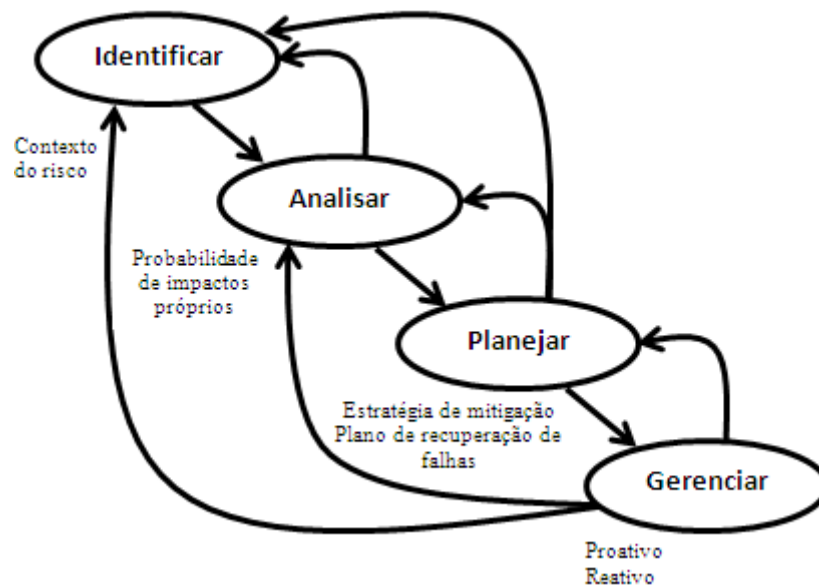


Figura 3 - Os 4 estágios do gerenciamento de projetos de desenvolvimento de software.
Fonte: Adaptado de Xiaosong (2009, p. 1).

Pressman (1996) enfatiza que a engenharia de software deve ser capaz de produzir modelos destinados à melhoria de processos com meio de aumentar a probabilidade de sucesso na execução de projetos de software e qualidade alcançada. Na mesma linha de Dey, Kinch e Ogunlana (2007), esse autor aponta que projetos de software apresentam altas taxas de desvios de prazo, custos e de qualidade. Estas diversas variáveis, muitas vezes são ignoradas pelos profissionais desenvolvedores de software. Assim, a necessidade de uso de metodologias orientadas ao processo de gerenciamento de riscos aumenta dentro das organizações visando assegurar a definição e cumprimento de metas custos, prazos e qualidade do produto gerado.

No entanto, em artigo recente apresentado na *International Conference on Computer Application and System Modeling*, Jun-guang e Zhen-Chao (2010), ao discutirem o uso de métodos de gerenciamento de riscos em projetos de software afirmam explicitamente que “... os métodos tradicionais de gerenciamento de projetos efetuam o gerenciamento do risco do projeto, mas não há nenhum projeto de pesquisa específico, ou um método específico de gestão de projetos que efetue a gestão de risco tratando o risco de software” (JUN-GUANG; ZHEN-CHAO, 2010 p. 8). Ainda, segundo esses autores, “pesquisas apontam que a engenharia de software moderna reconhece a importância do risco do projeto de software, mas de acordo com a probabilidade estatística, não é possível dar alguns modelos completos de gerenciamento de risco” (JUN-GUANG; ZHEN-CHAO, 2010 p. 8).

Ao discutir os riscos de projeto de software, Gusmão e Moura (2002), por exemplo, identificam três classes de risco: (i) *riscos de projeto de software*; (ii) *riscos de processos de software*; (iii) *riscos de produtos de software*. Os riscos de projeto de software dependem de parâmetros operacionais, organizacionais e contratuais associados ao projeto de desenvolvimento de software. Tal classe de riscos inclui a presença de restrições de recursos, existência de múltiplas interfaces, forma de relacionamento com fornecedores e ou presença de restrições contratuais (falta ou incompatibilidade de software, problemas de comunicação provenientes da distribuição geográfica dos recursos do projeto e indisponibilidade de agenda de recursos especialistas no processo de levantamento e desenvolvimento de requisitos críticos). Já os riscos de processo de software abrangem os riscos de projeto de software e os chamados riscos técnicos. Nos procedimentos de gerência esses riscos estão relacionados com o processo de planejamento, definição e contratação de equipe de trabalho, aceitação de compromissos relacionados à garantia de segurança (tais riscos podem estar contidos no processo de obtenção de recursos humanos, na garantia de qualidade, na decodificação e nos testes do software). Assim, riscos de processo emergem durante o levantamento e análise de requisitos, codificação e testes. Finalmente, os riscos de produto de software estão diretamente ligados às características intermediárias e finais do produto. Estes tipos de riscos têm origens nos requisitos de estabilidade do produto, desempenho, complexidade de codificação, especificação e aplicação de testes sobre o produto e suas partes (tais riscos estão associados às características dos requisitos, o nível de complexidade que foram especificados, testados e implementados).

Para Capers (1996, *apud* Schmitzetal 2006, p. 7), as características de um projeto bem-sucedido são: término no prazo, dentro do orçamento, alto nível de qualidade e de satisfação do usuário. Simetricamente, um projeto é considerado um fracasso quando: é cancelado antecipadamente, ultrapassa prazo ou orçamento previsto, apresenta baixa qualidade ou obtém um baixo nível de satisfação do usuário. Segundo Tao (2008), ao abordar projetos de desenvolvimento de software, o sucesso dos empreendimentos não pode ser alcançado sem a ocorrência de riscos e esses estão relacionados à possibilidade de falhas no atendimento do escopo do produto ou do projeto, qualidade insatisfatória, desvios nos custos e ou conclusão atrasada. Na mesma linha de argumentação, Dey, Kinch e Ogunlana (2007) salientam ainda que o sucesso do desenvolvimento de softwares depende dos seguintes critérios: qualidade (o software deve estar em conformidade com os requisitos solicitados pelo cliente), funcionalidade (o software deve executar todas as funcionalidades solicitadas pelo

cliente) e oportunidade (o software deve ser construído no tempo adequado em relação aos concorrentes). Cada um desses critérios é influenciado por fatores de risco específicos. No entanto, o grau de risco varia de acordo com a complexidade e tamanho dos empreendimentos de software sendo desenvolvidos.

Na indústria de software, a falta de compreensão e ou ambiguidades na definição de requisitos, escassez de recursos (financeiros, humanos, etc.), de hardware, de infraestrutura e de especificação de critérios relativos à segurança compreendem elementos de risco comuns ao desenvolvimento de softwares. Já Kwak e Stoddardb (2004) definem que falhas de projeto como o resultado da multiplicidade de riscos inerentes ao ambiente de desenvolvimento do software. Para esses autores e em muitas situações, projetos de desenvolvimento de software são abordados como conjunto de programas, com diversas interações e interdependências entre si. Tal fato conduz a incrementos na complexidade e nos riscos dos empreendimentos.

Também em muitas situações os projetos de desenvolvimento de software envolvem um processo de criação de algo que nunca foi feito antes, mesmo em casos de desenvolvimento de software similar. Como resultado da novidade e da complexidade esses projetos podem apresentar grande variabilidade nos custos, desvios no cronograma e problemas de qualidade e uso de software. Dessa forma, esses projetos apresentam grande carga de complexidade, urgência e novidade sendo bastante aderentes ao modelo de tipificação de projetos proposto por Shenhar e Dvir (2010).

Assim, definir e garantir o sucesso de um projeto é extremamente difícil. O escopo do projeto pode ser alterado devido a emergência de forças de mercado (marcadas por disponibilidade e nível de preço de insumos), necessidade de mudanças nos requisitos, disponibilidade de acesso a recursos humanos especializados que, em razão da dinâmica do setor, são constantemente redistribuídos entre os diversos *players* que atuam em um dado segmento. Adicionalmente, altas taxas de rotatividade de profissionais qualificados são apontadas por Kwak e Stoddardb (2004) como importante fator de risco. Segundo esses autores, gerentes de projetos de software na Índia perceberam que a rotatividade de recursos humanos como a sua maior fonte de risco em seus projetos, comprometendo escopo, prazo e custos.

Dey, Kinch e Ogunlana (2007) apresentam outra tipologia de riscos de projetos de software. Segundo esses autores os riscos podem ser classificados como *risco de mercado* (oriundo de flutuações na atividade econômica que tendem a afetar as condições de oferta e demanda de recursos e rentabilidade dos empreendimentos), *risco financeiro* (por meio de

flutuações no custo de capital, restrições de acesso ao financiamento de ativos de capital e capital de giro) e *risco técnico* (falhas no planejamento incluindo as associadas às estimativas e programação de emprego de recursos ou resultantes de desvios orçamentários, no cronograma, entre outros).

No que tange ao risco técnico, além dos desafios de gerenciamento de escopo, integração e qualidade, projetos de *software* demandam grande esforço no gerenciamento de tempo e custo, os quais se associam as decisões relacionadas quanto a alocação de recursos e sua produtividade. Segundo Dey, Kinch e Ogunlana (2007), apesar de alguns gestores alegarem que empregam técnicas de gerenciamento em seus projetos, há evidências de não se efetuar o gerenciamento de riscos de forma sistemática. Esse autor também salienta que em muitas situações os gestores desconhecem os riscos técnicos e que raramente lidam com os riscos financeiros e de mercado; fatores cruciais para se obter sucesso no desenvolvimento de *software*. Em suma, devido a sua criticidade riscos de mercado, financeiro e técnico estão diretamente ligados ao sucesso do projeto e não devem ser negligenciados pelo gerente e equipe alocada no projeto de desenvolvimento.

Tecnicamente, Xiaosong *et. al.* (2009) definem risco de desenvolvimento de *software* com um evento ou condição incerta com consequências negativas no projeto. Esse seria composto por três fatores: *causa raiz* (associada ocorrência de evento futuro que quando gerenciado de forma adequada pode minimizar o impacto de uma consequência indesejada ou otimizar o efeito de um evento desejado); *probabilidade* (por meio da qual se procura determinar a chance de ocorrência das fontes de causa raiz); *resultado* (efeito de uma ocorrência de uma ou mais causas raiz sobre o resultado do projeto).

De forma análoga, Tao (2008) define o risco como a possibilidade de má qualidade da solução de *software*. Essa implicaria em incremento nos custos, falha ou a conclusão atrasada do projeto, visto que o prazo inicial pode ser comprometido pela má alocação de recursos especialistas, do tamanho e complexidade do projeto não serem mapeadas de forma adequada na fase de planejamento e levantamento do escopo.

Dessa maneira e segundo Leme (2007), o sucesso de muitos projetos de *software* depende dos seguintes fatores: qualidade do planejamento; coordenação de equipes multidisciplinares; utilização de conhecimentos técnicos especializados; emprego de serviços de consultores externos; algum nível de terceirização de serviços, a depender da complexidade, urgência e ou novidade do projeto. Desta forma, na medida em que a escala dos empreendimentos, a complexidade, urgência e novidade (SHENHAR; DVIR, 2010) no

mercado de desenvolvimento de software crescem, também aumenta a necessidade da utilização de metodologias customizadas para o gerenciamento de riscos por parte de projetistas e gerentes de projetos de tecnologia da informação. Essas se orientam no sentido de garantir o atendimento de requisitos exigidos pelo cliente, dentro de padrões aceitáveis de custos e de prazo (GUSMÃO; MOURA, 2002).

Dey, Kinch e Ogunlana (2007) ressaltam que os riscos são parte integrante dos projetos de desenvolvimento de software. Como a fase de planejamento do projeto é geralmente feita com o mínimo da informação, o grau de risco tende a variar em função da complexidade, tamanho (tanto em termos de cronograma e orçamento). Alguns dos elementos de risco comuns associados a projetos de desenvolvimento de software são: falta de compreensão do escopo traduzida em requisitos ambíguos; escassez ou mal uso de equipamento de hardware; falta de definição dos requisitos ligados a aspectos de segurança; falta ou má alocação de recursos (incluindo recursos humanos especializados).

Em muitos casos, esses fatores estão ligados à causa raiz (XIAOSONG, *et al.* 2009), ao risco técnico (DEY; KINCH; OGUNLANA, 2007; KWAK; STODDARDB, (2004); emprego de mecanismos de gestão inadequados (TAO, 2008). Por isso, existe uma necessidade de gerenciar riscos internos associados aos projetos de desenvolvimento de software. A rigor, os riscos internos se associam a fatores que afetam direta e indiretamente o orçamento, o cronograma, a elaboração dos trabalhos, a capacitação do pessoal, a disponibilidade de recursos, as expectativas do(s) cliente(s) e os requisitos do projeto. Muitos desses possuem relação com o fato de as empresas ligadas ao setor atuarem em ambientes multi-projeto em meio a escassez de recursos humanos especializados e sem disporem de mecanismos customizados, destinados a promover a alocação desses em diferentes projetos de forma eficiente.

2.4 PROGRAMAÇÃO E ALOCAÇÃO DE RECURSOS EM PROJETOS

Em se tratando de alocação e planejamento de recursos, Mendes (2003) define que as ferramentas utilizadas para o sequenciamento de múltiplos projetos, como o gráfico de Gantt, o CPM (*Critical Path Method*) e o método PERT (*Program Evaluation and Review Technique*) podem apresentar limitações quando aplicadas a mais de um projeto, pois assumem que os recursos disponíveis são ilimitados e estes, aplicados a somente um projeto.

Mendes (2003) ainda define que um dos principais benefícios em se utilizar de forma otimizada os recursos por um determinado período de tempo em um projeto é a liberação do recurso o mais rápido possível para alocação em outros projetos da carteira e redução dos riscos de não cumprimento dos prazos previamente estabelecidos. Neste cenário, o grande desafio é definir a relação de precedência entre as atividades dos projetos e o tempo total de espera de alocação de um determinado recurso, pois existe, em um ambiente de múltiplos projetos, a restrição de recursos e a necessidade de redução de prazo das atividades.

Em relação aos riscos e desafios associados à necessidade de balanceamento de recursos especialistas entre os múltiplos projetos, Laslo (2010) aponta que o grande problema é estabelecer uma regra que permita que a organização opere diferentes projetos em um ambiente de escassez dos recursos especialistas e demandas trazidas pela estratégia da empresa.

Na verdade, a disputa pela atribuição de recursos especialistas em ambientes que envolvam múltiplos projetos é apontada por Laslo (2010) como uma fragilidade operacional que conduz a ineficiências. Dentre essas, tal disputa induz e caracteriza um dos principais motivos de conflitos/desacordos induzindo à intensificação de *lobbies* internos dentro das empresas. Assim, as atividades de planejamento e o escalonamento de recursos especialista são apontados por Laslo (2010) como críticas no gerenciamento de múltiplos projetos (ou GPP).

Os aspectos apontados acima remetem a necessidade de elaboração de técnicas destinadas a identificar e avaliar as possíveis variáveis estratégicas em termos de postura futura dos projetos das empresas em relação às limitações dos recursos especialistas existentes. Com o objetivo de minimizar tais problemas relacionados aos recursos especialistas, Laslo (2010) aponta que deverá ser feita a melhor otimização e alocação destes recursos, por meio de transferências desses recursos de seus departamentos para determinados projetos por um período de tempo pré-determinado, obtendo assim melhores resultados.

Com efeito, os aspectos apresentadas nas seções anteriores remetem à emergência de emprego de critérios destinados a alocação de recursos escassos (ou RE) em ambientes multi-projetos, fato que impacta diretamente a definição de regras relacionadas à definição de atributos associados às atividades da carteira de projetos (ou seja, interfere diretamente nos processos de GPP).

2.4.1 RECURSOS HUMANOS ESPECIALISTAS

De acordo com Machlup (1962) e Druker (1999), o conhecimento utilizado pelas organizações é produzido por profissionais, apontados como trabalhadores de conhecimento, que através de suas competências individuais para produzirem sistemas de gestão, de origem tecnológica ou estratégica. Trabalhadores de conhecimento dentro das empresas possuem um papel de grande importância para viabilizar a criação e a transferência de conhecimento, e são definidos por Boff e Abel (2005) como profissionais que transformam informação em insumo, combinando-a com seu conhecimento individual com o objetivo de produzir nova informação como resultado de sua atividade. A partir dessas características é possível determinar dentro das empresas, profissionais de conhecimento levando-se em consideração diversos níveis de especialização. Assim, torna-se necessário um melhor entendimento das características desse tipo de profissional, procurando identificar quais competências o diferenciam de um trabalhador tradicional e quais recursos por eles são utilizados para potencializar sua escala produtiva.

De certo modo, a identificação dos profissionais de conhecimento dentro das empresas pode ser considerada fácil. Mas um conceito mais profundo desse profissional é apresentado por Collins (1993), que atribui ao profissional de conhecimento a execução de atividades complexas e a utilização de recursos tecnológicos como apoio em suas tarefas. Cortada (1998) assevera que existem três características determinantes, presentes nos trabalhadores de conhecimento: (i) utilização e a produção de informação credenciando este profissional como um trabalhador especialista, (ii) a ampliação de sua produção intelectual e a sua melhora de desempenho, (iii) a proporção dentro das empresas é proporcional a complexidade das atividades.

O conhecimento, interno ou externo, do profissional do conhecimento consiste em teorias sobre as informações de importância para a realização de seu trabalho. Tais informações cessadas na memória do profissional na condução e execução das tarefas, acumuladas através de conhecimento e experiência são denominadas conhecimento interno, enquanto que as informações adquiridas com outros profissionais, livros, entre outros são denominados conhecimento externo.

Estes conhecimentos utilizam algum tipo de modelo mental (modelo mental é uma representação da realidade, organizada e indexada para que seja facilmente visualizada durante a execução de uma atividade/tarefa. Norman (1983) aponta que os profissionais de

conhecimento geralmente necessitam de um modelo mental de toda a situação antes que a tarefa seja executada, utilizando-se de insumos internos e externos para sua elaboração, sendo aperfeiçoado pelo processamento das informações e de acordo com sua experiência) e requer um alto nível de atenção do dos profissionais de conhecimento.

2.4.2 CLASSIFICAÇÃO DE RECURSOS HUMANOS ESPECIALISTAS

O trabalho de conhecimento é definido pela variedade ao invés da rotina de execução das atividades, pela exigência de profissionais de alto nível de habilidade e experiência, Boff e Abel (2005). Esta definição permite apontar que em uma mesma área de atuação existem dois tipos de profissionais: profissionais de conhecimento e profissionais tradicionais. Profissionais tradicionais executam suas atividades de forma rotineira e previsível, enquanto que profissionais de conhecimento são capazes de interpretar e processar as informações de suas tarefas em determinadas situações, inclusive situações críticas, produzindo novas informações para resolução de problemas e aumento de experiência. Os profissionais de conhecimento possuem um papel de destaque dentro das empresas, atuando em diversos cargos.

Tais papéis são classificados por Boff e Abel (2005) através dos profissionais que geram conhecimento - Quadro 3.

Quadro 3 - Classificação dos profissionais que criam conhecimento.

Criadores de conhecimento	Atividades	Exemplos de trabalhadores
Operadores	Acumulam e geram conhecimento tácito; habilidades adquiridas com a experiência	Profissionais da linha de frente
Especialistas	Acumulam, geram e atualizam conhecimento explícito	Cientistas, planejadores, analistas
Engenheiros	Transformam conhecimento tácito em explícito e vice-versa	Gerentes de nível médio
Gerentes	Gerenciam todo o processo de criação de conhecimento na empresa	Gerentes de nível sênior

Fonte: Adaptado de Boff e Abel (2005, p. 3).

Boff e Abel (2005) apontam que as atividades executadas por cada desses profissionais e as informações produzidas são de grande importância no processo de criação de conhecimento. A classificação apresentada no Quadro 4 é de grande valia, diferenciando os

profissionais através dos cargos que possuem, não levando em conta suas competências. O nível de experiência de cada profissional deve ser considerado dentro das empresas, devido a sua importância e na capacidade de cada um aplicá-la adequadamente. Em relação a profissionais especialistas, Boff e Abel (2005) realçam a dependência da empresa deste tipo de recurso em relação aos demais profissionais, pois profissionais especialistas possuem uma concentração maior de conhecimento em um determinado indivíduo que será utilizado de forma potencializada na realização de tarefas estratégicas da empresa.

Profissional especialista é definido por Abel (2001) como o indivíduo que possui habilidades cognitivas com o objetivo de resolver problemas estratégicos com qualidade e com desempenho superiores aos demais profissionais. Este autor também aponta outra característica deste profissional é o fato de usar conhecimento como insumo principal na execução de tarefas, obtendo assim, melhor desempenho. Assim, o profissional especialista se destaca dos demais profissionais pelo fato de utilizar maior quantidade de conhecimento como insumo no processamento das atividades – demonstrado na Figura 4, além de possuir a capacidade de aplicar tais informações na resolução de determinado problema.

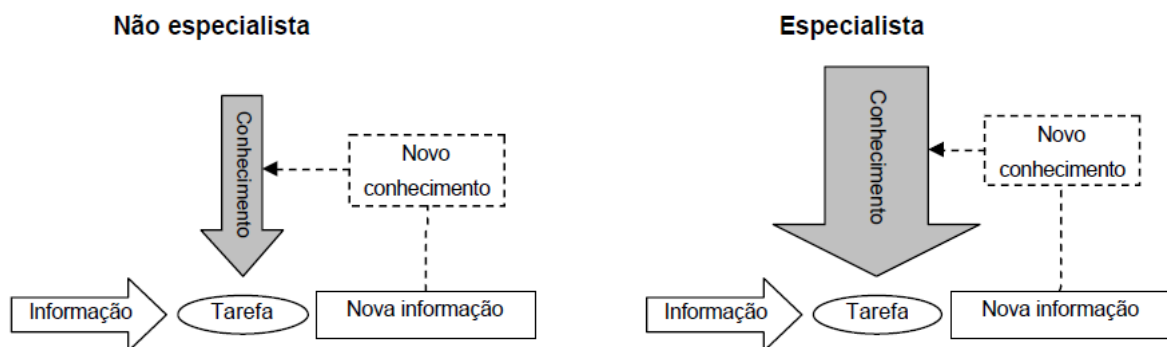


Figura 4 – Diferença de conhecimento utilizado entre um profissional especialista e um não especialista.

Fonte: Boff e Abel (2005, p. 3).

Na metodologia, Abel (2001), Vanlehn (1989) e Ericsson e Smith (1991), definem que o desempenho de um profissional especialista pode ser aferido através de determinadas variáveis que indiquem sua perícia. Uma variável de grande importância é o tempo de treinamento, onde um profissional especialista pode ser definido por uma pessoa que efetuou mais de 20.000 horas em treinamento e na solução de problemas críticos nos últimos 10 anos. A perícia por si só não é o único indicador de um especialista (Figura 5). Nesse sentido, Ericsson e Smith (1991) apresentam que a partir da carga horária efetuada em treinamentos,

os indivíduos ampliam seu desempenho em função da prática estabelecida. Os autores também apontam que independente do desenvolvimento das perícias dos indivíduos, a união de talento com oportunidade de desenvolvimento intelectual do profissional aumenta a possibilidade de atingir níveis de desempenho maiores.

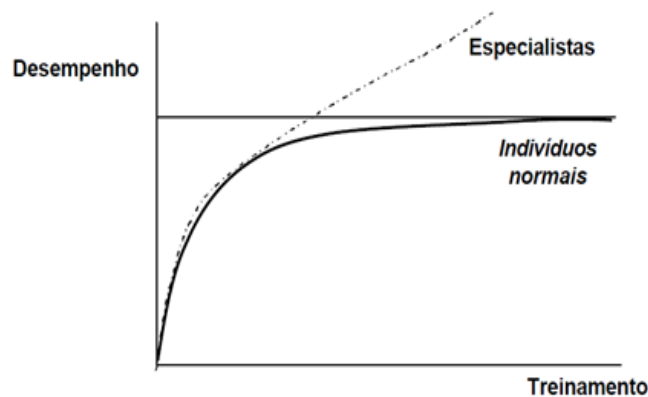


Figura 5 – Representação aproximada da curva de aprendizado de indivíduos submetidos a treinamento comparado com a de especialistas.

Fonte: Boff e Abel (2005, p. 7).

Outra medida de desempenho de profissionais especialista é exposta por Turban (1983), onde um especialista atinge uma medida de desempenho a partir do número de problemas solucionados em relação ao número de problemas apresentados maior (em torno de 3 vezes maior que a média) que o número dos profissionais dentro de um mesmo departamento. Por fim, Boff e Abel (2005) demonstram que, em se tratando de uma tarefa crítica, em média 10% de todos os profissionais possuem desempenho maior que o índice de 85% de problemas críticos resolvidos com sucesso, contra o índice de 30% dos demais profissionais, demonstrado na Figura 6.

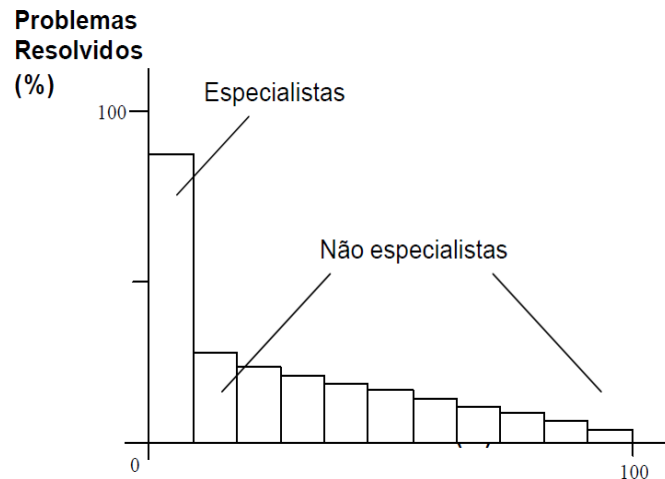


Figura 6 – Desempenho comparativo entre especialistas e não especialistas em um determinado domínio.

Fonte: Boff e Abel (2005, p. 8).

2.4.3 RECURSOS HUMANOS ESPECIALISTAS NO AMBIENTE DE DESENVOLVIMENTO DE SOFTWARE

Em ambientes caracterizados pela produção sob encomenda, foco no desenvolvimento e com ênfase no ciclo de vida do produto, Pressman (2006), problemas relacionados ao sequenciamento das tarefas podem ocorrer. De acordo com Muniz (2009), sequenciar atividades significa efetuar a ordenação das tarefas com o objetivo de maximizar ou minimizar uma determinada função relacionada ao custo ou ao tempo da produção. Geralmente esses problemas envolvem atividades que devem ser completadas, levando-se em consideração metas (em relação a prazo) e, às vezes, em relação a custos. Baker (1974) aponta que problemas relacionados ao escalonamento de tarefas são aqueles que possuem alocação de recursos especialistas no tempo com o objetivo de executar um conjunto de tarefas. Um destes problemas, dentro de um ambiente de produção, é conhecido como *job shop problem – JSP*.

De acordo com Muniz (2009), o JSP consiste em se determinar um sequenciamento ideal para se processar n Jobs em n máquinas, maximizando ou minimizando restrições relativas a custo (como o makesplan – tempo total decorrido), ao flow time (tempo de espera e processamento) ou de lateness (atraso máximo). De acordo com Galway (2004) e Daneshkhah (2004), as tarefas podem ser particionadas em atividades que necessitam de serem executadas por determinados recursos em uma determinada ordem. De acordo com Pacheco e Santoro

(1999), ambientes caracterizados por problemas de JSP são considerados grandes desafios para serem solucionados computacionalmente, além de não existir um algoritmo matemático eficiente capaz de garantir a solução ideal. Assim, Fattahi *et. al.* (2006) assevera que os problemas de JSP são considerados ainda um dos mais difíceis problemas de otimização combinatória.

2.5 O MODELO CLÁSSICO DE JOB SHOP PROBLEM

O problema clássico de JSP é caracterizado por Guimarães (2007) como um determinado conjunto J de Jobs (trabalhos, atividades) e um determinado número M de Máquinas determinadas para execução dos Jobs. No modelo clássico de job shop há um dado conjunto J de trabalhos (jobs) e um conjunto M de máquinas. Cada trabalho $j \in J$ consiste em um conjunto de operações $jjk \in O_{j \in M}$ que $j \in J, k \in K : \{1, 2, \dots, n\}$ $n_{jj} \in \mathbb{N}$ e n é o número total de operações para cada trabalho j , e $jk \in \mu$ é o número da máquina para processar a k -ésima operação do trabalho j , ou seja, $\{m_1, m_2, \dots, m_{jk}\} \subseteq \mu \in M$. Todas as máquinas são diferentes e suas velocidades de processamento são constantes.

O ambiente de JSP clássico é marcado por um conjunto de restrições como: (i) um único trabalho dever ser escalonado uma única vez; (ii) cada trabalho pode ser escalonado em qualquer máquina; (iii) uma máquina poderá executar um único trabalho por vez; (iv) uma máquina só poderá ser liberada quando a tarefa atual for finalizada e (v) somente máquina de mesmo tipo de operação. A formulação do problema de JSP contido nas restrições acima é representada por $n \times |M|$ para todo j . A Figura 7 representa um problema de JSP de dimensão 3×4 (3 máquinas e 4 tarefas – jobs), onde cada valor corresponde na tabela ao tempo da atividade de cada job em cada máquina.

		M_1	M_2	M_3	M_4
J_1	$O_{1,1}$	1	4	5	8
	$O_{2,1}$	7	5	6	5
J_2	$O_{1,2}$	2	5	6	2
J_3	$O_{1,3}$	12	5	4	7
	$O_{2,3}$	5	6	3	5
	$O_{3,3}$	2	4	12	5

Figura 7 – Job shop clássico 3x4.
Fonte: Adaptado de Guimarães (2007, p. 66).

A solução para este problema pode ser representada através da utilização de uma lista de sequenciamento das atividades, respeitando as regras de formulação de JSP, apresentado na Figura 8.

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
(1,1,4,1)	(1,2,2,1)	(2,2,1,2)	(2,1,4,1)	(3,2,2,2)	(3,1,1,3)	(1,3,3,3)	(2,3,4,1)

Figura 8 – Solução ótima para um job shop clássico 3x4.
Fonte: Adaptado de Guimarães (2007, p. 66).

2.5.1.1 TIPOS DE MODELOS PROPOSTOS PARA RESOLUÇÃO DE JSP

Na busca de uma solução ótima para o problema de JSP, alguns modelos matemáticos são apresentados. De acordo com Lagewag, Lenstra e Rinnoy Kan (1979), o JSP é considerado um problema de fator combinatorial e de complexidade NP-completo. Assim, Pacheco e Santoro (1999) apontam que os modelos compostos por algoritmos otimizantes são considerados computacionalmente viáveis quando estes são aplicados a problemas pequenos. Já para problemas considerados de grande porte, às vezes, pode-se penalizar a obtenção da melhor solução por uma solução de caráter subótimo através do uso de métodos heurísticos, obtendo como resultado um tempo computacional aceitável. Por este motivo, Pacheco e Santoro (1999) apresentam os modelos matemáticos para resolução de JSP, baseado em teste de hipóteses, em dois conjuntos: conjuntos de solução ótima (testes paramétricos) e conjuntos de soluções

heurísticas (testes não paramétricos). A classificação hierarquizada para a solução de JSP é demonstrada na Figura 9. De acordo com Siegel e Castellan (2006), os testes de hipótese são estabelecidos através de um conjunto de dados. Para isso, é necessário possuir um procedimento para aceitar ou rejeitar a hipótese em questão. Siegel e Castellan (2006) apontam também que testes de hipóteses se dividem em testes paramétricos e testes não paramétricos. Testes paramétricos são aqueles que utilizam os parâmetros da distribuição, ou valores estimados destes para estabelecer o cálculo de sua estatística. Na maioria das vezes, estes testes são mais rigorosos e possuem mais pressuposições para sua validação. Em contrapartida, os testes não paramétricos utilizam, para o cálculo de sua estatística, postos atribuídos aos dados ordenados e são livres da distribuição de probabilidades dos valores em questão. Segundo Callegari-Jacques (2003), testes paramétricos são caracterizados pelo fato da variável estudada ter distribuição normal ou aproximação normal. Em contrapartida, testes não paramétricos, não têm exigências quanto ao conhecimento da distribuição da variável na população.

Por fim, Shimakura (2012) conclui que testes paramétricos são mais rigorosos e robustos em relação aos testes não paramétricos, devido ao fato de possuírem mais pressuposições para a sua validação, enquanto que os testes não paramétricos possuem para a composição da sua estatística, pontos atribuídos por dados ordenados livres da faixa da distribuição das probabilidades dos dados em questão.

A seguir serão descritos os modelos de soluções ótimas, soluções ótimas baseadas em programação dinâmica, modelos de soluções heurísticas e, por fim, soluções heurísticas baseados em algoritmos genéticos.

2.5.1.2 MODELOS DE SOLUÇÕES ÓTIMAS

Pacheco e Santoro (1999) definem os modelos de soluções ótimas como modelos marcados pelo uso de técnicas genéricas na busca de soluções para problemas combinatoriais, exceto os modelos que utilizam algoritmos eficientes. Algoritmo eficiente é denominado por Muniz (2009) como o algoritmo que possui o tempo decorrido em busca da solução ótima pode ser escrito através de uma função polinomial em relação ao tamanho do problema.

As técnicas genéricas utilizadas são consideradas procedimentos enumerativos que não abordam ou utilizam de alguma característica de informação específica do problema em prol da busca da solução do problema. Estes modelos, em geral, podem ser utilizados na busca de solução de problemas de escalonamento (*scheduling*), sendo diferenciados pela definição das restrições, parâmetros e o seu objetivo. Para solução dos problemas de JSP, o tempo computacional para a utilização de algoritmos exatos em busca da solução ótima cresce exponencialmente em relação ao crescimento do tamanho da entrada do problema (JOHNSON, 1967 *apud* VASCONCELOS, 2007). Desse modo, resolver problemas mais complexos pode inviabilizar o uso de abordagem computacional. Métodos exatos escolhem a solução ótima através do uso de regras simples. Estas são aplicadas sobre o conjunto de valores que possuem todas as soluções. Esse tipo de método, utilizado para problemas específicos, pode reduzir o impacto causado através do volume de soluções, visto que o algoritmo se aproveita de particularidades para apresentar a solução.

Os problemas ocasionados pelo JSP são considerados NP, por este motivo, soluções através de métodos exatos podem ser inviáveis computacionalmente (JOHNSON, 1967 *apud* VASCONCELOS, 2007). Alguns algoritmos que apresentam uma porcentagem alta para a solução ótima do problema, onde seu tempo computacional varia de forma polinomial em relação ao tamanho do problema.

2.5.1.3 MODELOS DE SOLUÇÕES ÓTIMAS BASEADOS EM PROGRAMAÇÃO DINÂMICA

De acordo com Ferreira (2005), Programação Dinâmica (PD) é um método iterativo que geralmente é utilizado para determinar problemas ligados à otimização. Pacheco e Santoro (1999) definem que o princípio da otimização é o alicerce da programação dinâmica.

O método de PD trabalha através do processo de decomposição de um problema em subproblemas menores. Os subproblemas são solucionados e os resultados parciais são armazenados para então serem combinados com os sobressaltados dos subproblemas menores para resolver problemas maiores, até ocorrer à resolução do problema original. Modelos de PD são aplicáveis à problemas de otimização. Em contrapartida, soluções consideradas não ótimas de subproblemas devem ser descartadas, tornando os métodos de PD mais eficientes que os demais algoritmos. Segundo Ferreira (2005), os subproblemas são resolvidos uma única vez e suas soluções armazenadas podem ser utilizadas inúmeras vezes.

Devido ao grande volume de processamento realizado na busca de uma solução ótima, dentro de um ambiente de software, Ferreira (2005) declara que os algoritmos de PD demandam grande espaço de armazenamento dos dados além de gastar tempo excessivo no processo de execução do algoritmo. Ferreira (2005) anota que o espaço necessário para o armazenamento da matriz total de um algoritmo PD pode ultrapassar os limites do hardware (máquina) em processamento, onde o tempo necessário para execução do algoritmo pode ser sobrecarregado pela falta de eficiência do uso de memória da máquina. Assim, esta discussão é asseverada por Ferreira (2005), indicando que o tempo de processamento (execução) dos algoritmos de PD em determinados hardwares (máquinas) pode se tornar um processo irrealizável, de acordo com a quantidade de sequências a serem executadas.

O modelo proposto por Laslo (2010) é situado dentro do contexto da hierarquia dos modelos de soluções ótimas para a resolução para o problema de Job Shop Problem, como método de programação dinâmica, baseado pelo fato que: quando as variáveis do problema não estão totalmente envolvidas e a solução pode ser apresentada de modo eficiente tendo como base técnicas de otimização como a Programação Dinâmica, (BALLARD E BROWN, 1982).

2.5.1.4 MODELOS DE SOLUÇÕES HEURÍSTICAS

Os modelos que possuem suas soluções baseadas em heurística são definidos por Pacheco e Santoro (1999) como modelos que proporcionam uma boa solução para o problema e esta solução necessariamente não precisa ser uma solução ótima, mas ela deve ser alcançada em um tempo adequado. Assim, um modelo heurístico abdica da garantia de encontrar a melhor solução para que uma determinada solução possa ser encontrada em sua execução de

maneira rápida. Para isso, os modelos heurísticos podem utilizar duas estratégias: (i) utilização da heurística de passo único (construção da sequência e a programação da solução executados de uma única vez, sem considerar as alternativas possíveis), (ii) utilização da heurística de busca (exploração de um subconjunto de possíveis soluções para o problema). Tal maneira, os modelos heurísticos de passo único como os modelos heurísticos de busca possuem capacidade de centralizar a sequência das operações dos problemas ou nos gargalos atuais existentes.

Em relação a sua complexidade, modelos heurísticos são elaborados quando o uso de equações matemáticas torna-se impraticável ou quando utilizadas de forma dispendiosa. Estes modelos são baseados em regras empíricas (intuitivas) e, em determinada solução, permitem o avanço para outra solução mais aperfeiçoada. São considerados modelos de busca inteligentes em relação ao processo de tomada de decisão e de acordo Pacheco e Santoro (1999), tais modelos são elaborados tendo como base técnicas de inteligência artificial.

2.5.1.5 MODELOS DE SOLUÇÕES HEURÍSTICAS BASEADOS EM ALGORITMOS GENÉTICOS

De acordo com Pacheco e Santoro (1999), modelos que geram soluções ótimas, com exceção dos que possuem algoritmos eficientes, caracterizam-se pelo uso de técnicas genéricas para solução de problemas de utilização de unidades de tempo. Essas técnicas genéricas são procedimentos enumerativos que não utilizam informação ou conhecimento de características específicas do problema para direcionar a busca. Tais modelos podem ser utilizados nos problemas de scheduling, diferindo pela forma como são definidas a função objetivo, as restrições e os parâmetros. Dentre os modelos de solução ótima enumerativos, a enumeração pode se dar explicitamente (todas as alternativas tem o resultado calculado) ou implicitamente (algumas combinações não são exploradas por possuírem resultado parcial deduzido como inferior a outro já obtido). Gonçalves, Mendes e Resende (2004) classificam os algoritmos genéticos como algoritmos adaptativos, que podem ser utilizados para resolverem problemas relacionados à otimização de operações. Os algoritmos genéticos se baseiam em princípios de seleção natural para acercar uma gama de problemas. Pacheco e Santoro (1999) classificam estes algoritmos como robustos e de fácil adaptação, baseado em técnicas utilizadas em diversas áreas. Se forem bem codificadas, estes algoritmos são capazes de progredir as soluções para os problemas do mundo real, incluindo os problemas de

otimização. Quanto à estrutura e funcionamento de algoritmos genéticos, Pacheco e Santoro (1999) definem que o princípio básico do funcionamento dos algoritmos genéticos é que um critério de seleção vai fazer com que, depois de muitas gerações, o conjunto inicial de indivíduos gere indivíduos mais aptos. A maioria dos métodos de seleção é designada para escolher preferencialmente indivíduos com maiores notas de aptidão, embora não exclusivamente, a fim de manter a diversidade da população.

Um método de seleção muito utilizado é o Método da Roleta, onde indivíduos de uma geração são escolhidos para fazer parte da próxima geração, através de um sorteio de roleta – Figura 10.

Algoritmo Genético

```

{
    Gerar população inicial  $P_t$ 
    Avaliar população  $P_t$ 
    Enquanto critério de parada não satisfeito Repetir
    {
        Selecionar elementos de  $P_t$  para copiar em  $P_{t+1}$ 
        Cruzar elementos de  $P_t$  e colocar em  $P_{t+1}$ 
        Transformar elementos de  $P_t$  e colocar em  $P_{t+1}$ 
        Avaliar nova população  $P_{t+1}$ 
         $P_t = P_{t+1}$ 
    }
}

```

Figura 10 – Exemplo de algoritmo genético.

Fonte: Adaptado de Gonçalves, Mendes e Resende (2002, p. 5).

2.5.1.6 MODELO PROPOSTO POR LASLO (2010)

O modelo proposto por Laslo (2010) é um modelo não paramétrico para cálculo da melhor solução ótima para o problema de JSP baseado no processo de otimização e alocação de recursos humanos especialistas. Para isso utiliza transferências desses recursos de seus departamentos para determinados projetos por um período de tempo pré-determinado, obtendo assim melhores resultados. O modelo de Laslo (2010) é um modelo matemático não paramétrico para cálculo da melhor solução ótima para o problema de JSP baseado no processo de otimização e alocação de recursos humanos especialistas. O modelo de Laslo

(2010) faz o tratamento da minimização dos custos das penalidades contratuais. Para isso, baseia-se na informação da contratação de recursos especialistas entre diferentes projetos de um portfólio.

2.5.1.7 MODELO PROPOSTO POR GONÇALVES, MENDES E RESENDE (2004)

Gonçalves, Mendes e Resende (2004) apresentam um modelo para solução do JSP, baseado em algoritmo genético, utilizando o conceito de chaves aleatórias para o tratamento de tempo das operações (atividades). O modelo utiliza um algoritmo genético para a definição de prioridades das operações através da parametrização dos prazos das atividades. Como forma de saída, o modelo apresenta um cronograma, com o objetivo de buscar a melhor solução ótima para o problema.

Os autores utilizaram em seu modelo o conceito de programação ativa com o objetivo de controlar estes atrasos e diminuir o tempo conclusão da operação, Gonçalves e Beirão (1999).

O modelo proposto por Gonçalves, Mendes e Resende (2004) é o resultado da combinação entre o uso de algoritmo genético, um processo que calcula o tempo ativo das operações e de uma funcionalidade que efetua a busca local das operações, respeitando as seguintes fases: (i) atribuição das prioridades das operações, (ii) processo de construção e (iii) procedimento de busca local (efetua a análise do resultado da programação ativa, visando melhorar a solução obtida).

5.3 Modelo proposto por Müller, Rodrigues e Gómez (2006)

O modelo proposto por Müller, Rodrigues e Gómez (2006) utiliza o makesplan, o tempo total de paradas e o tempo total de atrasos para a solução de JSP clássico. O modelo aborda dois problemas associados ao JSP: (i) seleção das partes (para o processo de geração de famílias, o modelo se baseou no método de análise por fluxo de produção proposto por Kusiak and Chow (1987) e (ii) escalonamento das partes (a finalidade é efetuar o melhor sequenciamento das atividades nas máquinas disponíveis por meio de percurso pré-definido em função do tempo). O processo de sequenciamento é feito de através de métodos de otimização (um dos algoritmos em destaque para este método são os algoritmos genéticos).

Müller, Rodrigues e Gómez (2006) ressaltam que além de ter como objetivo a otimização do makesplan, o modelo efetua a gestão das datas de entrega (minimização do atraso) e a produtividade (minimizar as paradas de produção).

3 METODOLOGIA

3.1 TIPO DE PESQUISA

De acordo com Acevedo e Nohara (2006), Cooper e Schindler (2008), Gil (2006), Vergara (1988), o tipo de pesquisa exploratório tem como objetivo promover uma aproximação ou entendimento inicial de um dado tema. Tal estratégia deve viabilizar uma melhor compreensão do problema a ser estudado, entendimento de suas variáveis e inter-relacionamento entre essas, decomposição e ou estratificação de fases necessárias para sua solução, bem como o levantamento e análise de informações disponíveis associadas a um problema. Nesse tipo de investigação, a pesquisa parte de certas premissas, onde tais informações são geralmente originadas por meio de estudos bibliográficos e ou levantamento de dados por meio de entrevistas. De acordo com Severino (2007), a pesquisa exploratória tem como principal objetivo identificar as informações referentes a um determinado assunto, delineando o campo de estudo e identificando as condições do objeto em estudo.

3.2 PESQUISA QUALITATIVA

De um modo geral, a pesquisa qualitativa constata a evidência ou não de algum fenômeno, sem se importar com sua intensidade ou amplitude. É denominada qualitativa em contrariedade à pesquisa quantitativa, em função da forma como os dados serão tratados e da forma de percepção de uma realidade, em que, no caso da pesquisa qualitativa, o mundo é conhecido por meio de experiência e senso comum (conhecimento intuitivo), em oposição às abstrações (modelos) da pesquisa quantitativa.

Uma pesquisa qualitativa costuma ser direcionada e não tem como finalidade medir determinado evento através de métodos estatísticos, mas obter dados descritivos através de contato direto entre o pesquisador e o cenário de estudo em questão. Na pesquisa qualitativa, o pesquisador procura compreender os fenômenos em relação a perspectiva dos envolvidos no cenário em estudo, permitindo assim sua interpretação, estimulando os entrevistados envolvidos a exporem seus pensamentos sobre um determinado assunto.

Segundo Severino (2007), a pesquisa qualitativa possui caráter exploratório, tendo como estímulo fazer com que os entrevistados passem a pensar livremente sobre algum tema,

objeto ou conceito. Martins e Theóphilo (2007) demonstram que os aspectos subjetivos e atingem motivações não explícitas, ou mesmo conscientes, de maneira espontânea. Pode ser utilizada quando se procura obter percepções e entendimento sobre a natureza geral de uma questão, abrindo espaço para a interpretação.

Martins e Theóphilo (2007) ainda definem que a pesquisa qualitativa é uma pesquisa indutiva, onde o pesquisador desenvolve conceitos e entendimentos a partir de padrões encontrados nos dados, ao invés de coletar dados para comprovar teorias, hipóteses e modelos pré-concebidos. Para Godoy (1995), um estudo qualitativo pode ser conduzido por diferentes caminhos, onde os mais utilizados e conhecidos são: a pesquisa documental e o estudo de caso. A pesquisa documental representa uma maneira de aprofundar nos estudos de novos temas, visto que os documentos ganham papel de destaque, pois geralmente são consideradas poderosas fontes de pesquisa para estudos qualitativos.

Godoy (1995) determina que os documentos primários (como jornais, revistas, obras científicas e técnicas) são produzidos por pessoas que experimentaram e viveram o assunto sendo estudado, e por secundários quando os documentos não foram escritos por pessoas presentes ao assunto.

3.3 ESTRATÉGIA DE PESQUISA

Para este estudo, foram adotadas a pesquisa bibliográfica e a análise documental como estratégia de pesquisa. A pesquisa bibliográfica é fundamental e obrigatória em qualquer modalidade de pesquisa. De modo geral, qualquer informação publicada (impressa ou eletrônica) pode se tornar uma fonte de consulta.

Martins e Theóphilo (2007) apontam que os produzidos se convertem em fontes dos temas a serem pesquisados, onde o pesquisador efetua seu trabalho a partir das contribuições dos autores dos estudos analíticos constante dos textos.

Já a etapa de análise dos documentos tem como objetivo produzir conhecimento e criar novas formas de compreender os fenômenos. Segundo Severino (2007), o pesquisador deve compreender os dados dos textos produzidos e condensar as informações, pode delimitar as tendências e na medida do possível, fazer a conclusão. Martins e Theóphilo (2007) definem que os documentos não existem isoladamente, mas precisam ser localizados em uma estrutura teórica para que o seu conteúdo seja compreendido. Assim, é essencial efetuar em todas as

etapas de uma análise documental uma avaliação do contexto histórico no qual foi produzido o documento, o contexto sócio-político do autor.

3.4 TÉCNICA DE PESQUISA

As técnicas de pesquisa são consideradas comportamentos operacionais que ajudam como mediação prática para a execução de pesquisas, podendo ser utilizadas em pesquisas realizadas com a ajuda de metodologias distintas. Para isso, precisa ser conciliáveis com a metodologia adotada, respeitando a natureza e o limite do conhecimento (SEVERINO, 2007).

A entrevista é caracterizada como uma técnica que possibilita estabelecer o relacionamento entre o entrevistador e o entrevistado. Segundo Martins (2007), a entrevista pode ser considerada uma conversa orientada com a finalidade de obter informações e dados para a pesquisa, resultantes da aplicação de um questionário.

Um questionário é caracterizado por Severino (2007) como um bloco de questões elaborado de forma sistemática que tem o objetivo de obter informações de maneira escrita das pessoas pesquisadas, visando conhecer a opinião do entrevistado em relação ao assunto estudado. Para a elaboração de um questionário, Severino (2007) assevera que as questões devem ser objetivas, com a determinação de motivarem respostas justamente objetivas, de modo a evitar dúvidas, respostas duplicadas e concisas.

3.5 O ESTUDO DE CASO

O estudo de caso como ferramenta de investigação científica é utilizado para compreender processos na complexidade social nas quais estes se manifestam, tanto em situações problemáticas, para análise dos obstáculos, ou em situações bem-sucedidas, para avaliação de modelos exemplares (YIN, 2010). De acordo com Severino (2007), estudo de caso trata-se de uma modalidade de pesquisa muito específica, pois consiste no estudo profundo e exaustivo de um único objeto ou de poucos objetos (um caso particular), dependendo fortemente do contexto do estudo, e seus resultados não podem ser generalizados. Para Yin (2010), estudo de caso é considerado uma inquisição de origem empírica com a finalidade de investigar um fenômeno dentro de um contexto real.

Yin (2010) define que a metodologia de estudo de caso pressupõe, em alguns casos, a existência de uma teoria prévia, que será testada no decorrer da investigação, e admite em outros casos a construção de uma teoria a partir dos achados da pesquisa. Yin (2010) afirma que o estudo de caso pode ser usado, pelo menos, em cinco situações: (i) para explicar vínculos causais em intervenções na vida real que são muito complexas para estratégias experimentais, (ii) quando é preciso descrever intervenções no contexto em que ocorrem, (iii) para ilustrar determinados tópicos em uma investigação, (iv) para explorar uma situação complexa de resultados e (v) como uma forma de meta-avaliação de determinados processos.

Ao apontar os métodos para a elaboração de uma pesquisa, Yin (2010) define projeto de pesquisa como uma sequência lógica que conecta os dados coletados de maneira empírica às questões iniciais do estudo e as suas conclusões, que se remete a quatro problemas: (i) quais questões devem ser estudadas, (ii) quais dados são relevantes, (iii) quais dados devem ser coletados e (iv) como devem ser analisados os resultados. O estudo de caso, de acordo com Yin (2010) adquire evidências a partir das seguintes fontes de dados: documentos, registros de arquivos, observação, entrevistas e artefatos físicos.

Yin (2010) enfatiza que a entrevista é uma das mais importantes fontes de dados para a elaboração de um estudo de caso. Neste contexto, uma entrevista é considerada uma fonte crucial de evidências para o estudo de caso. Segundo Yin (2010), dois pontos devem ser considerados: (i) manter um alinhamento de investigação (refletido no estudo de caso) e, (ii) formulação de questões verdadeiras e de maneira imparcial adequada à linha de investigação.

Visto que um estudo de caso ocorre no ambiente real do próprio caso, Yin (2010) aponta que diante deste cenário, é possível a criação da observação direta, uma vez que se acredita que os fenômenos não tenham apenas registros históricos, mas que possuam condições ambientais e comportamentos significativos para realizar a observação.

Severino (2007) aponta que o caso escolhido para pesquisa deve ser relevante e figurado a ponto de fundamentar uma generalização para situações análogas, permitindo inferências. Yin (2010) assevera a discussão apontando que os dados devem ser coletados e registrados com a maior exatidão possível, devendo ser trabalhados mediante uma rígida análise e demonstrados de maneira qualificada.

3.6 A ESCOLHA DO MÉTODO

Este estudo é de caráter qualitativo, uma vez que se pretendeu analisar a perspectiva do JSP com restrição de RHE em uma empresa de desenvolvimento de softwares.

Como estratégia de pesquisa foi adotada a pesquisa bibliográfica e a análise documental, pois de acordo com Martins e Theóphilo (2007), a pesquisa bibliográfica é realizada a partir da anotação de pesquisas anteriores impressas, como teses, artigos, revistas, etc., sendo baseada em categorias (teóricas ou de dados) já trabalhada por outros pesquisadores, devidamente registrada. Já Yin (2010) aponta que a análise documental é uma importante fonte de pesquisa oriunda de diversas formas, como cartas, memorandos, atas de reuniões, documentos administrativos e estudos formais, com o objetivo de ajudar a estabelecer com nitidez a empresa em estudo a partir da qualidade dos documentos coletados.

Segundo Vergara (1998), uma pesquisa pode ser classificada quanto aos seus objetivos e meios de investigação. Neste sentido, quanto à natureza, optou-se pela pesquisa exploratória, pois se pretendeu, com seus resultados, um entendimento melhor do JSP no contexto de projetos de desenvolvimento de softwares.

Já o estudo de caso é apontado por Godoy (1995) como um exame apurado de um ambiente, de um sujeito ou de um determinado cenário e possui a finalidade de conceder o comportamento da realidade por meio da discussão, análise e o ensaio de um determinado assunto ou problema real. O fato de o estudo de caso ser uma das estratégias mais utilizadas pelos pesquisadores é enfatizado Godoy (1995), devido à dificuldade de controle e de apontar respostas a questões “como” e “por quê” sobre determinados fenômenos que ocorrem em um cenário real. Segundo Yin (2010), as observações diretas podem ser realizadas por meio de visita de campo ou por coleta de dados resultante de entrevistas.

Diante deste contexto, para este estudo, foi adotado o estudo de caso como ferramenta de investigação científica e como técnica de pesquisa foi adota a entevista semi-estruturada como instrumento de apoio a elaboração do estudo de caso.

Este trabalho não tem o propósito de eleger um modelo para resolução de JSP superior ao outro. Para essa conclusão, é interessante a realização de pesquisas com uso de simulações numéricas a fim de comparar os resultados gerados para cada modelo.

3.7 DELINEAMENTO DA PESQUISA

A escolha do método de pesquisa deste trabalho foi determinada para que os resultados obtidos pela pesquisa fossem direcionados para a realização de seus objetivos. Para isso, o fluxo da elaboração desta pesquisa respeitou os seguintes critérios:

- a) Estudo de três modelos matemáticos utilizados para a resolução do JSP;
- b) Avaliação do comportamento de cada modelo em relação a diversos critérios de aderência previamente estabelecidos para sua aplicação em um ambiente de desenvolvimento de software;
- c) Escolha dos modelos matemáticos baseados em diferentes algoritmos (modelos baseados em solução ótima – modelos paramétricos e modelos heurísticos – modelos não paramétricos);
- d) Avaliação da aderência de cada modelo através de um estudo de caso em uma empresa de desenvolvimento de software.

Para este estudo, foram adotados os modelos propostos por Laslo (2010), Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006). Para a escolha dos modelos, foram observadas as características de cada modelo em relação ao método matemático utilizado para o processo de resolução do JSP, o comportamento em modelos de JSP até e superiores a 10x10 (10 recursos x 10 atividades), o tratamento do modelo em relação aos custos ocasionados por desvios de prazo, a possibilidade de priorização de atividades.

3.8 PROCEDIMENTOS METODOLÓGICOS

A questão de pesquisa descrita nesse trabalho é abordada por meio de avaliação a aderência dos modelos de otimização para o problema de Job Shop Problem – JSP proposto por Laslo (2010), Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) em um ambiente projetizado de desenvolvimento de software, conforme procedimento explicitado a seguir.

Num primeiro momento foi realizada a descrição dos modelos de otimização proposto por Laslo (2010), Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) para o entendimento dos parâmetros de cada modelo e sua contribuição para a resolução do problema de Job Shop Problem.

Para a identificação e avaliação das contribuições dos modelos de Laslo (2010), Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) para o estudo, serão criados os seguintes indicadores de aderência de cada modelo a partir de requisitos previamente estabelecidos:

- (i) dimensão (que avalia o modelo através de do tamanho da dimensão do modelo);
- (ii) custos (finalidade de estimar o tratamento do custo por atividade e em relação ao custo total do projeto);
- (iii) escalonamento (com a finalidade de avaliar o tratamento do modelo em relação à priorização da data de entrega do projeto e o atraso das atividades ou atraso total do projeto);
- (iv) características (determinam o tipo de modelo matemático usado e seu comportamento em relação ao JSP e estimam a utilização do modelo em um ambiente de desenvolvimento de software) com o objetivo de avaliar o comportamento e a eficácia do modelo dentro de um ambiente de desenvolvimento de software;
- (v) resumo (apresenta as características ideais do uso do modelo em relação ao JSP).

Para a validação dos resultados obtidos por meio da avaliação dos requisitos descritos anteriormente, será avaliada a aderência do uso do modelo proposto Laslo (2010), Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) em um estudo de caso, em uma empresa de desenvolvimento de desenvolvimento de software. Será realizada uma entrevista semi-estruturada com o gestor de projetos da empresa, de acordo com a seguinte estrutura:

- a) Qual a média anual da alocação de recursos (humanos e máquinas) nos projetos de sua competência?
- b) Como são tratados os atrasos em relação aos projetos?
- c) Em caso de desvio de prazo de entrega do projeto, quais são os fatores críticos que são analisados?
- d) Como é feita a alocação de recursos humanos especialistas nos projetos, principalmente para projetos com alto nível de criticidade?
- e) A priorização da data de entrega dos projetos é uma prática frequente dentro da carteira de projetos?
- f) Como um modelo matemático poderia ajudar no processo de alocação de recursos em projetos?

A empresa estudada é uma companhia líder no mercado de idioma hispano-português de desenvolvimento de softwares para Contact Center. Iniciou suas atividades no Brasil no

estado de São Paulo, no ano de 1999. Atualmente a empresa é considerada uma das maiores empregadoras privadas no território brasileiro além de ser uma das mais importantes formadoras de mão-de-obra qualificada no Brasil.

3.9 LIMITAÇÕES DA PESQUISA

Essa pesquisa tem como objetivo descrever os modelos matemáticos de algoritmo genético e de otimização e suas respectivas vantagens e desvantagens para resolução de *Job Shop Problem* em projetos de desenvolvimento de software.

4 RESULTADOS E DISCUSSÃO

Os problemas relacionados a escalonamento, segundo Baker (1974), são aqueles que estão relacionados à alocação de recursos no tempo com o objetivo de executar um conjunto de tarefas além de suas soluções serem consideradas complexas e de grandes desafios computacionais. Neste contexto, um recurso é considerado uma máquina e uma tarefa é denominada job e cada job tem seu início e término bem definidos. No geral, problemas de escalonamento estimam um conjunto de tarefas (atividades) que devem ser executadas em um determinado conjunto de máquinas com a finalidade de efetuar a melhor distribuição destas atividades às máquinas visando aperfeiçoar o desempenho, caracterizando assim o JSP. Laslo (2010), Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) apresentam modelos matemáticos em busca da solução dos problemas de JSP. Laslo (2010) apresenta um modelo matemático de otimização não paramétrico, enquanto Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) apresentam seus modelos baseados em algoritmos genéticos. Este trabalho procurou avaliar as contribuições de cada modelo na busca da solução dos problemas causados pelo JSP, dentro de uma empresa de desenvolvimento de softwares com utilização de RHE.

4.1 MODELO PROPOSTO POR LASLO (2010)

O modelo proposto por Laslo (2010) é um modelo não paramétrico para cálculo da melhor solução ótima para o problema de JSP baseado no processo de otimização e alocação de recursos humanos especialistas. Para isso utiliza transferências desses recursos de seus departamentos para determinados projetos por um período de tempo pré-determinado, obtendo assim melhores resultados.

4.1.1 APRESENTAÇÃO DO MODELO

O modelo de Laslo (2010) faz o tratamento da minimização de penalidades contratuais. Para isso, baseia-se na informação da contratação de recursos especialistas entre diferentes projetos de um portfólio.

A definição de variáveis associadas ao início, precedências, duração e probabilidade de finalização do projeto e portfólio de projetos utilizados pelo modelo apresentado por Laslo (2010) são apresentadas no Quadro 4 e suas respectivas notações estão relacionadas nos Quadros 5, 6, 7 e 8. Embora o número de variáveis seja expressivo, essas são aderentes com a rede Pert e ao método CPM tradicional e permitem a descrição de todos os aspectos do projeto relacionados à duração (estimada e efetiva).

Quadro 4 - Notação e definição de variáveis associadas ao início, precedências, duração e probabilidade de finalização do projeto e portfólio de projetos. (*)

ID	Notação	Definição e descrição da variável
1A	R_k	Momento que o k-ésimo projeto é aceito. O tempo que o h-ésimo projeto é aceito $1 \leq k \leq m$ (pré-determinado).
2A	G_k	Trata-se de um Partial Ordered Set - POSET ou Conjunto Parcialmente Ordenado e inicial das atividades dos projetos. Para cada k-ésimo projeto, esse descreve o Direct Acyclic Graph – DAG inicial de atividades sendo $1 \leq k \leq m$ (onde m = número de projetos).
3A	G_{k_t}	Ponto no tempo em que se obtém a atualização do POSET das atividades dos projetos. DAG que descreve no tempo t o up-to-date POSET das atividades do k-ésimo projeto $A_{k,i,j} R_k \leq t < F_{k,i,j}$.
5A	D_k	Tempo contratual para conclusão do projeto. Tempo contratual de conclusão do k-ésimo projeto $1 \leq k \leq m$ (pré-determinado).
6A	F_k	Tempo de conclusão de todos os projetos. Tempo de conclusão real do k-ésimo projeto $F_k = \text{Max}_i(F_{k,i,q_k})$.
7A	F	Tempo de conclusão de todas as tarefas. Tempo final do Makesplan $F = \text{Max}_k(F_k)$.
8A	P_k^*	Probabilidade de o projeto ser executado antes do tempo contratual estipulado. A probabilidade de confiança exigida (restrição chance) do k-ésimo projeto ser realizado antes ou no momento de conclusão contratual D_k (pré-determinado).

Fonte: Laslo (2010, p. 612).

(*) O makesplan corresponde ao momento no tempo em que a execução de todas as atividades associadas a um projeto é finalizada. O makesplan compreende o intervalo de tempo entre o início e fim de um projeto.

O Quadro 5 relaciona os parâmetros associados ao enquadramento das atividades nos diferentes projetos do portfólio, os nós de início e término e a duração estocástica das diferentes atividades.

Quadro 5 - Notação e definição de variáveis associadas às atividades, momento de início e término nos nós, ao início, relação de precedência, duração e probabilidade de finalização do projeto. (*)

ID	Notação	Definição e descrição da variável
1B	$A_{k,i,j}$	Atividade executada no período. Atividade de k-ésimo projeto que pode ser executada de forma contínua no período compreendido entre as ocorrências de eventos (nós) i e j, $1 \leq i \leq q_k - 1$, $2 \leq j \leq q_k$, $A_{k,i,j} \in G_k$ (q_k – número de nós em DAG G_k).
2B	$S_{k,i,j}$	Tempo que a atividade realmente se inicia. Tempo em que a atividade $A_{k,i,j}$ realmente começa (valor aleatório).
3B	$F_{k,i,j}$	Tempo real de conclusão de cada tarefa por projeto O tempo em atividade $A_{k,i,j}$ é realmente realizado, $F_{k,i,j} = S_{k,i,j} + t_{k,i,j}$ (valor aleatório).
4B	t	Ponto de decisão do Makesplan Ponto de decisão no agendamento de tempo dentro do makesplan [0,F]
5b	$t_{k,i,j}$	Duração de cada tarefa de cada projeto Duração aleatória de $A_{k,i,j}$ com valor esperado $E(t_{k,i,j})$ e variância $V(t_{k,i,j})$.

Fonte: Laslo (2010, p. 612).

(*) O makesplan corresponde ao momento no tempo em que a execução de todas as atividades associadas a um projeto é finalizada. O makesplan compreende o intervalo de tempo entre o início e fim de um projeto.

O Quadro 6 apresenta os parâmetros relacionados ao emprego de recursos especializados em cada projeto do portfólio bem como o tempo previsto de contratação (ou alocação do respectivo recurso nos diferentes projetos), data de liberação e custo direto associado ao emprego do recurso por período de tempo.

Quadro 6 - Notação e definição dos parâmetros de tipo, custo e momento de contratação e liberação associados ao emprego de recursos especialistas.

ID	Notação	Definição e descrição da variável
1C	I_h	Número de especialistas. O h-ésimo especialista $1 \leq h \leq n$ (n – número de especialistas).

2C	$n_{k,i,j}$	Indica o índice do especialista para execução da atividade. Índice do especialista individual para executar a atividade de projeto $A_{k,i,j}$, $1 \leq n_{k,i,j} \leq n$ (pré-determinado).
3C	T_h^*	Tempo previsto de contratação do especialista. O tempo previsto da contratação de especialistas individuais I_h (uma variável ideal, a ser determinada antecipadamente).
4C	T_h^{**}	Tempo de liberação do especialista. Tempo de liberação do especialista individual I_h (valor aleatório).
5C	C_h^I	Custo do especialista por período de trabalho Custo do salário de cada especialista I_h por unidade de tempo dentro do período $[T_h^*, T_h^{**}]$ (pré-determinado), $1 \leq h \leq n$.

Fonte: Laslo (2010, p. 612).

O Quadro 7 apresenta as variáveis associados aos custos fixos e variáveis relacionadas às penalidades caso o projeto termine após a data de término contratualmente estabelecida.

Quadro 7 - Notação e definição dos parâmetros associados à penalidade fixa e variável em função do atraso no k-ésimo projeto do portfólio .

ID	Notação	Definição e descrição da variável
1D	C_k^d	Custo da penalidade por não conclusão do projeto no prazo contratual. Componente fixo da função de custo da pena por não completar o k-ésimo projeto no tempo $D_k, F_k > D_k$ (pré-determinado).
2D	C_k^v	Custo da penalidade por unidade de tempo de trabalho Componente variável da função de custo da pena por unidade de tempo de atraso dentro do prazo $[D_k, F_k]$ se $F_k > D_k$ (pré-determinado).

Fonte: Laslo (2010, p. 612).

Finalmente, o Quadro 8 apresenta as variáveis associados aos custos totais associados à ociosidade dos recursos especialistas, custo associado à penalidade por atrasos e gasto total relacionado a essas variáveis em termos do portfólio do projeto.

Quadro 8 - Notação e definição do custo total fixo e variável do atraso e emprego de recursos especialistas. (*)

ID	Notação	Definição e descrição da variável
----	---------	-----------------------------------

1E	Z	Despesas pendentes contidas no Makesplan. Planejamento / programação de despesas dependentes, ou seja, despesas de dependentes de T_h^* dentro do Makesplan $[0,F]$, $Z=Z_1+Z_2$
2E	Z_1	Custo ocioso do especialista dentro do Makesplan. Custo dos salários ociosos dos especialistas individuais dentro do makesplan $[0,F]$
3E	Z_2	Custo da pena do atraso das despesas. Pena de despesas de atraso.

Fonte: Laslo (2010, p. 612).

(*) O makesplan corresponde ao momento no tempo em que a execução de todas as atividades associadas a um projeto é finalizada. O makesplan compreende o intervalo de tempo entre o início e fim de um projeto.

Dadas essas variáveis, Laslo (2010) propõe um modelo de otimização heurística que também utiliza procedimentos de simulação computacional para a solução do JSP no contexto de GPP.

Na verdade trata-se de um modelo de minimização baseado nas equações (1), (2), (3), (4) e (5). A rigor, o modelo está orientado para a minimização do valor esperado com as despesas $Z = Z_1 + Z_2$ associados à definição de T^*h (tempo de contratação do recurso especialista).

O exercício de minimização está sujeito a aspectos bem conhecidos na literatura de gerenciamento de prazo em projetos associados as datas de início e término de cada atividade. Assim, as atividades não se iniciam antes: do momento R_k de o projeto ser aceito (equação 2); do término de todas as atividades antecessoras (equação 3); antes do recursos especialista executar a atividade sob sua responsabilidade (equação 4). Adicionalmente, um mesmo recurso especialista não pode estar alocado em mais de uma atividade no mesmo momento no tempo (equação 5). Finalmente, a equação 6 indica a chance de isso ocorrer. Então o modelo proposto por Laslo (2010) apresenta um modelo de minimização de custo, tendo como base a definição de T^*h condicionado as seguintes restrições: início do projeto, término de atividades antecessoras, conclusão de atividades dos recursos especialistas, execução simultânea de atividades pelo recurso especialista antes do makesplan.

$$E Z = \text{Min}_{\{T_h^*\}} [E Z_1 + E Z_2] \quad \text{Eq. (1)}$$

$$S_{(k,i,j) \in G_k} \geq R_k, 1 \leq k \leq m \quad \text{Eq. (2)}$$

$$S_{(k,i,x,j) \in G_k} \geq \text{Max}_i F_{(k,i,j=x) \in G_k}, 2 \leq x \leq q_k - 1 \quad \text{Eq. (3)}$$

$$S_{(k,i,j) \in G_k} \geq T_h^* \text{ se } n_{k,i,j} = h, 1 \leq k \leq m, 1 \leq h \leq n \quad \text{Eq. (4)}$$

$$\left[S_{z,x,y}, F_{z,x,y} \right] I \left[S_{(k,i,j)l(z,x,y)}, F_{(k,i,j)l(z,x,y)} \right] = \theta \text{ se } n_{z,x,y} = n_{k,i,j} \quad \text{Eq. (5)}$$

Fica implícito que aplicação do modelo de minimização proposto por Laslo (2010), sujeito as três ordens de restrição apresentadas acima, se baseia na construção de planilhas com os parâmetros dos projetos (momento de início, relação de atividades, relações de precedência, duração, etc.), sua implementação por meio de funções que assegurem a verificação das restrições (2) e (3). Já as restrições (3) e (4) são construídas por meio de funções lógicas, sendo o critério de alocação dos recursos especialistas, feito após a implantação de um processo de simulação, definido na equação (6).

$$\Pr(F_k \leq D_k) \geq p_k^*, \quad 1 \leq k \leq m \quad \text{Eq. (6)}$$

Associadas diretamente ao problema de minimização apresentado sumariamente acima, as equações (7), (8), (9), (10) e (11) apresentadas abaixo representam, respectivamente: o custo relacionado ao tempo de fluxo ou *lead time* ou ainda o custo da ociosidade dos especialistas, uma vez dadas às condições (3), (4) e (5), pode haver janelas de tempo ocioso na programação; o momento aleatório no tempo onde o recurso especialista individual I_h será liberado; a probabilidade de que isso aconteça; o dispêndio associado à penalidade por atraso; o termo binário que capta a existência ou não desse atraso.

$$Z1 = \sum_{h=1}^n C_h^I \left(T_h^{**} - T_h^* - \sum_{k=1}^m \sum_{i=1}^{q_k-1} \sum_{j=2}^{q_k} t_{k,i,j} \delta_{k,i,j}^h \right) \quad \text{Eq. (7)}$$

$$T_h^{**} = \text{Max}_{(k,i,j) \in G_k} F_{k,i,j} \delta_{k,i,j}^h, \quad 1 \leq k \leq m \quad \text{Eq. (8)}$$

$$\delta_{(k,i,j) \in G_k}^h = \begin{cases} 1 & \text{se } n_{(k,i,j) \in G_k} = h \\ 0 & \text{caso contrário} \end{cases} \quad 1 \leq h \leq n \quad \text{Eq. (9)}$$

$$Z_2 = \sum_{k=1}^m \gamma_k \left[C_k^d + C_k^v F_k - D_k \right] \quad \text{Eq. (10)}$$

$$\gamma_k = \begin{cases} 1 & \text{se } F_k > D_k \\ 0 & \text{caso contrário} \end{cases} \quad 1 \leq k \leq m \quad \text{Eq. (11)}$$

4.2 MODELO PROPOSTO POR GONÇALVES, MENDES E RESENDE (2004)

Gonçalves, Mendes e Resende (2004) apresentam um modelo para solução do JSP, baseado em algoritmo genético, utilizando o conceito de chaves aleatórias para o tratamento de tempo das operações (atividades). O modelo utiliza um algoritmo genético para a definição de prioridades das operações, parametrizando os tempos. Como forma de saída, o modelo apresenta um cronograma, com o objetivo de buscar a melhor solução ótima para o problema de JSP.

O modelo proposto pelos autores utiliza para a composição das precedências das operações o conceito de cronograma ativo. Este tipo de cronograma é caracterizado por possíveis horários em que nenhuma operação pode se inicializar sem causar penalidade ou atraso em outra operação, nem a quebra da precedência existente entre as duas operações. De acordo com Gonçalves e Beirão (1999), este tipo de programação é denominado programação ativa, sendo baseado em horários em que uma máquina fique ociosa o menor tempo possível ou que o tempo de ociosidade não ultrapasse um valor previamente definido, com a finalidade de se obter um cronograma sem atrasos. Gonçalves, Mendes e Resende (2004) apontam que a qualidade do makesplan pode ser afetada através de cronogramas muito grandes e com desvios de prazos entre as operações (atividades). Os autores utilizaram em seu modelo o conceito de programação ativa com o objetivo de controlar estes atrasos e diminuir o tempo conclusão da operação, Gonçalves e Beirão (1999).

O modelo proposto por Gonçalves, Mendes e Resende (2004) é o resultado da combinação entre o uso de algoritmo genético, um processo que calcula o tempo ativo das operações e de uma funcionalidade que efetua a busca local das operações, respeitando as seguintes fases: (i) atribuição das prioridades das operações (através do uso de um algoritmo genético com a finalidade de efetuar a prioridade e os possíveis atrasos das operações), (ii) processo de construção (esta fase faz uso das prioridades ou dos atrasos da fase anterior como parâmetros de entrada para a programação ativa) e (iii) procedimento de busca local (efetua a análise do resultado da programação ativa, visando melhorar a solução obtida).

4.2.1 APRESENTAÇÃO DO MODELO

O algoritmo genético proposto pelos autores utiliza o conceito de chave aleatória (utilização dos valores 0 ou 1) para efetuar o cruzamento de possíveis soluções viáveis ao problema de JSP, transpondo a grande parte do problema para a função objetivo do algoritmo. Para a análise dos pontos de cruzamento em um algoritmo genético, Gonçalves, Mendes e Resende (2004) apontam que para um melhor desempenho, o algoritmo deve utilizar parâmetros uniformes do tipo crossovers (utilização de 2 possíveis pontos de solução para o problema). Levando-se em consideração que cada lado de uma moeda pode misturar características de cada um dos pontos do cruzamento crossover, respeitando um valor de probabilidade fornecido previamente de forma empírica. A Figura 11 apresenta um exemplo de uma distribuição crossover.

Lançamento Moeda	H	H	T	H	T
Solução 1	0.57	0.93	0.36	0.12	0.78
Solução 2	0.46	0.35	0.59	0.89	0.23
Melhor Desempenho	0.57	0.93	0.59	0.12	0,23

Figura 11 – Exemplo de distribuição crossover uniforme.
Fonte: Adaptado de Gonçalves, Mendes e Resende (2002, p. 7).

Ao se confrontar com um ponto crossover de probabilidade pequena de solução do problema, este método efetua a criação de uma nova população (ponto de solução) de maneira

aleatória em conjunto com a população original e os dados de melhor desempenho, evitando assim a convergência de informações entre as populações, representado na Figura 12:

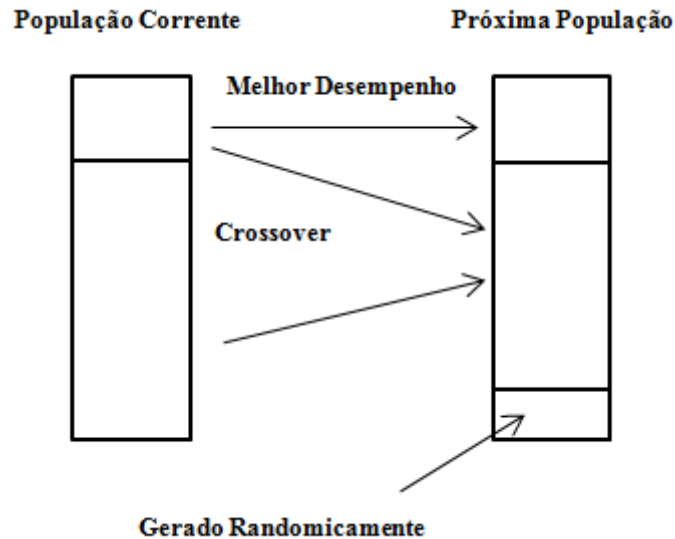


Figura 12 – Processo de geração de populações consecutivas.
Fonte: Adaptado de Gonçalves, Mendes e Resende (2002, p. 7).

4.2.2 PROCESSO DE CONSTRUÇÃO

O algoritmo genético utilizado para a construção da programação ativa tem como principal característica o incremento de tempo em cada ponto de solução. Para cada iteração g , existe um tempo de programação t_g , o conjunto de programação ativo abrange todas as operações em t_g é demonstrado pelas seguintes equações:

Eq. (12)

Eq. (13)

Eq. (14)

Eq. (15)

Eq. (16)

Sendo que:

g = iteração atual

t = tempo associado a iteração

J = número de Jobs (atividades)

F = tempo de chegada das atividades

d = duração das atividades

m = valor randômico gerado

S = operação agendada para a iteração g atual

E = operação precedente

A equação (12) define o conjunto ativo que compreende todas as operações que operam no modelo em t . A capacidade de máquinas restantes em t é apresentada na equação (13). A representa todas as operações agendadas até a iteração g e B compreende o tempo de término das operações em t . O tempo de atraso associado à iteração g é descrita pelas fórmulas (14) e (15). As operações que são precedências viáveis para o intervalo da programação estão descritas em (5).

O algoritmo genético responsável por gerar as programações de tempo ativo é demonstrado na Figura 13.

Inicialização: $g=0, t_g=0, A_0=\{0\}, RMC_m(0)=1, F_g(0)=\{0\}, S_g(0)=\{0\}$

Enquanto $|S_g| \leq n+1$ **repetir**

{

Incremento da iteração

$$g = g+1$$

Determina o tempo associado com a iteração g

$$t_g = \text{Min}_{j \in A_g} \{F_j\}$$

Cálculo $A_g(t_g), RMC_m(t_g), E_g = E_g(t_g, Delay_g)$

enquanto $E_g \neq \{\}$ **repetir**

{

Seleciona a operação de maior prioridade

$$j^* = \arg \max_{j \in E_g} \{PRIORIDADE_j\}$$

Cálculo do tempo de chegada mais cedo (em termos de precedência apenas)

$$EF_{j^*} = \max_{i \in P_j} \{F_i\} + d_{j^*}$$

Cálculo do tempo de chegada mais cedo (em termos de prioridade e capacidade)

$$F_{j^*} = \min \left\{ t \in [EF_{j^*} - d_{j^*}, \infty] \cap F_g \mid r_{j^*,m} \leq RMC_m(\tau), m \mid r_{j^*,m} > 0, \tau \in [t, t + d_{j^*}] \right\} + d_{j^*}$$

Incremento da iteração

$$g = g+1$$

Cálculo $A_g(t_g), RMC_m(t_g), E_g = E_g(t_g, Delay_g)$

Atualiza S_g e F_g

$$S_g = S_{g-1} \cup \{j^*\}$$

$$F_g = F_{g-1} \cup \{F_{j^*}\}$$

}

}

Cálculo do Makespan

$$F_{n+1} = \text{Max}_{i \in P_{n+1}} \{F_i\}$$

Figura 13 – Algoritmo usado para construção de parâmetros de tempo ativos.

Fonte: Adaptado de Gonçalves, Mendes e Resende (2002, p. 8).

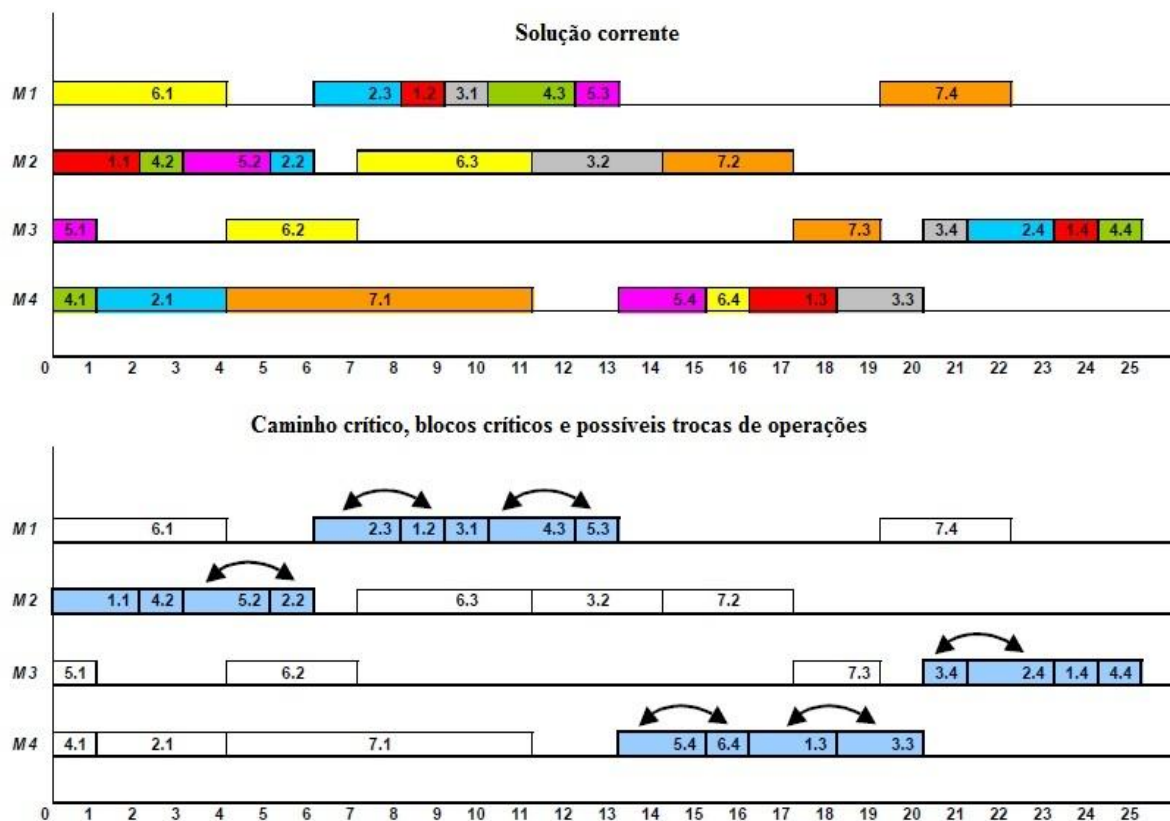
Desse modo, o makesplan da solução é determinado pelo número máximo de todas as operações antecessoras a operação.

4.2.3 PROCEDIMENTO DE BUSCA LOCAL

A primeira etapa desta busca é identificar o resultado obtido no processo de construção de programação ativa o caminho crítico da solução, onde qualquer operação contida é denominada operação crítica. Assim, é possível dissociar o caminho crítico encontrado em conjuntos.

Cada conjunto é definido como uma sequência máxima de operações críticas imediatas que estão requisitando a utilização da mesma máquina. A elaboração dos conjuntos é feita baseada na seguinte condição: caso uma operação e uma máquina antecessora for considerada operação crítica, será escolhida a operação antecessora que aparece primeiro na sequência do resultado da programação. Esta fase está baseada no algoritmo proposto por Nowicki and Smutnicki (1996) – Figura 14.

Desse modo, o caminho crítico é originado pela seguinte condição: Um conjunto B , onde $l < l < b$, deverá ser trocado as duas últimas e as duas primeiras operações do conjunto, caso contrário, onde $l=1$ (b), será trocado apenas o último e o primeiro conjunto de operações.



**Figura 14 – Algoritmo proposto por Nowicki and Smutnicki (1996).
Fonte: Adaptado de Gonçalves, Mendes e Resende (2002, p. 10).**

Para casos que o primeiro e/ou o último conjunto disporem de apenas duas operações, elas serão automaticamente trocadas. Em casos de um conjunto conter apenas uma operação, não haverá nenhuma troca. A troca só será concretizada caso houver melhora no makesplan, caso contrário, a troca é desfeita. Com a troca, o caminho crítico pode sofrer mudanças. Se o caminho crítico for alterado, um novo será traçado. Caso não houver mudanças no caminho crítico, o makesplan será afetado de forma positiva. A decisão da troca é feita através de um algoritmo genético demonstrado na Figura 15.

```

Busca Local (SoluçãoCorrente)
Faça
{
    AlteraçãoSoluçãoCorrente = Falso

    Determina o caminho crítico e os blocos críticos da SoluçãoCorrente

    enquanto blocos não processados e não AlteraçãoSoluçãoCorrente faça
    {
        se não primeiro bloco crítico então
            NovaSolução := troca as primeiras duas operações do bloco na
                SoluçãoCorrente

            se makespan(NovaSolução) < Makespan(SoluçãoCorrente) então
                SoluçãoCorrente = NovaSolução
                AlteraçãoSoluçãoCorrente = Verdadeiro
            Fim se
        Fim se

        se não último bloco crítico e não AlteraçãoSoluçãoCorrente então
            NovaSolução := troca as últimas duas operações do bloco na
                SoluçãoCorrente

            se makespan(NovaSolução) < Makespan(SoluçãoCorrente) então
                SoluçãoCorrente = NovaSolução
                AlteraçãoSoluçãoCorrente = Verdadeiro
            Fim se
        Fim se
    }
}
faça enquanto AlteraçãoSoluçãoCorrente = Falso

Retorno SoluçãoCorrente

```

Figura 15 – Algoritmo usado busca local.

Fonte: Adaptado de Gonçalves, Mendes e Resende (2002, p. 10-11).

4.3 MODELO PROPOSTO POR MÜLLER, RODRIGUES E GÓMEZ (2006)

O modelo proposto por Müller, Rodrigues e Gómez (2006) utiliza o makesplan, o tempo total de paradas e o tempo total de atrasos para a solução de JSP clássico. O modelo aborda dois problemas associados ao JSP: (i) seleção das partes (para o processo de geração de famílias, o modelo se baseou no método de análise por fluxo de produção proposto por Kusiak and Chow (1987)) e (ii) escalonamento das partes (a finalidade é efetuar o melhor sequenciamento das atividades (Jobs) nas máquinas disponíveis por meio de percurso pré-definido em função do tempo). O processo de sequenciamento é feito de através de métodos de otimização (um dos algoritmos em destaque para este método são os algoritmos genéticos).

De acordo Müller, Rodrigues e Gómez (2006), o escalonamento das partes é um problema complexo e de difícil resolução. Para resolver o problema, os autores utilizaram o algoritmo de meta-heurística Busca Tabu, que consiste em apresentar aproximações com o máximo de precisão para a solução ótima do problema de otimização. De acordo com Glover e Laguna (1997), o algoritmo de Busca Tabu possui 3 fundamentos básicos: (i) utilização de uma lista dos dados para manter o histórico da evolução do processo de busca, (ii) utilização de um mecanismo de controle para balancear a aceitação ou não de uma nova configuração, para isso, baseia-se nas informações de restrições e aspirações desejadas da lista e (iii) junção de comportamento que alternam as estratégias de diversificação e intensificação.

Müller, Rodrigues e Gómez (2006) ressaltam que além de ter como objetivo a otimização do makesplan, o modelo efetua a gestão das datas de entrega (minimização do atraso) e da produtividade (minimizar as paradas do prazo total contratual).

4.3.1 APRESENTAÇÃO DO MODELO

O modelo apresentado por Müller, Rodrigues e Gómez (2006) é baseado no modelo SMF (Sistema de Manufatura Flexível) proposto por Stecke (1986). O escalonamento das partes foi formalizado a partir das seguintes variáveis:

N = número de partes

I = índice para a parte

K = índice para a máquina

= data de entrega da parte

= data da saída de produção

P = família das partes

A formulação é apresentada pelas seguintes equações e restrições:

Eq. (17)

Eq. (18)

Eq. (19)

Eq. (20)

Eq. (21)

O modelo possui uma equação pela busca de minimização do problema de escalonamento (17) que utiliza as variáveis de tempo (e) e a representação das famílias de partes (p). O algoritmo se apoia em e variáveis para o processo de tomada de decisão: (i) o tempo total de produção representado pela equação (18) que contém o tempo inicial da

primeira atividade (job) até o tempo final da última atividade (job) processada, (ii) o tempo de atraso total é representado na equação (19) e (iii) tempo total de paradas entre 2 conjuntos de produção (20) e a restrição (21) certifica a não negatividade das variáveis de entrada do modelo. A partir da função objetivo (17) o algoritmo Busca Tabu foi modificado para resolver o problema de escalonamento causado pelo JSP, representado na Figura 16.

```

enquanto ( $niter - melhiter < nbmax$ ) faça
  para cada “m” máquina faça

     $f' \leftarrow f_{melhor}$ ;
     $niter \leftarrow melhiter + 1$ ;
     $p \leftarrow p^*$ 
    Gerar  $Vp_e^*$  de soluções  $(e, p^*)_i$  em  $N_e(e, p^*)$  ou  $f((e, p^*)_i)$ 
    Atualizar Lista Tabu L e A (z);

    se  $f(e', p^*) < f_{melhor}$  então
       $f_{melhor} \leftarrow f(e', p^*)$ ;
    fim se

    Gerar  $Vp_p^*$  de soluções  $(e, p)_i$  em  $N_p(e, p)$  ou  $f((e, p)_i) <$ 
    Atualizar Lista Tabu L e A (z);

    se  $f(e', p) < f_{melhor}$  então
       $f_{melhor} \leftarrow f(e', p)$ ;
    fim se

    Gerar  $Vp^*$  de soluções  $(e, p)_i$  em  $N_p(e, p)$  ou  $f((e, p)_i) < A(f(e, p))$ :
    Atualizar Lista Tabu L e A (z);

    se  $f(e', p) < f_{melhor}$  então
       $f_{melhor} \leftarrow f(e', p)$ ;
    fim se

     $p \leftarrow p'$ ;
     $e \leftarrow e'$ ;
    se  $f_{melhor} < f'$  então
       $melhiter \leftarrow niter$ ;
    fim se
fim enquanto

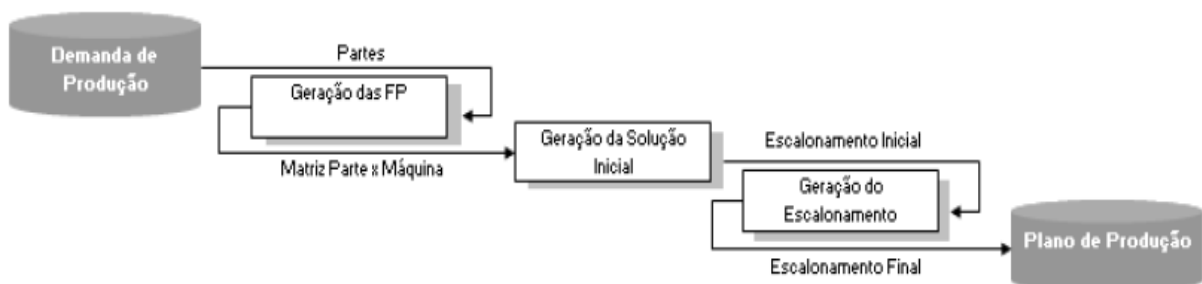
```

Figura 16 – Algoritmo Busca Tabu modificado.
Fonte: Adaptado de Müller, Rodrigues e Gómez (2006, p. 3).

Um novo conjunto é gerado a partir de uma solução ótima de cada iteração do modelo restrita pela lista Tabu. A função f é aperfeiçoada a cada iteração da função critério A (critério para aspiração). A parada do algoritmo pode ser controlada pela variável $nbmax$ (número máximo de iterações permitidas sem melhora na função f) ou executar o algoritmo até se seu valor mínimo seja alcançado.

4.3.2 ARQUITETURA DO MODELO

Para o processo de seleção e escalonamento de partes, os autores utilizaram 5 fases sequenciais – ver Figura 17. A primeira fase é responsável pela busca das informações técnicas e data de entrega armazenadas. Na segunda fase é aplicado o algoritmo de identificação de agrupamentos obtendo como resultado a matriz de máquina x parte. A terceira fase é responsável pela geração do escalonamento inicial (solução inicial viável independente, que obrigatoriamente deve fazer parte do conjunto de soluções possíveis da amostra). A partir do escalonamento inicial gerado na terceira fase é possível aplicar o algoritmo Busca Tabu modificado para a otimização do escalonamento - processo é realizado na quarta fase. Após a geração do escalonamento final, o resultado é armazenado pela quinta fase no plano de produção, de modo que seu histórico pode ser comparado com o escalonamento atual.



**Figura 17 – Arquitetura do modelo proposto por Müller, Rodrigues e Gómez (2006).
Fonte: Adaptado de Müller, Rodrigues e Gómez (2006, p. 4).**

4.4 CONTRIBUIÇÕES DOS MODELOS PARA O SOLUÇÃO DO JSP

A solução para os problemas de escalonamento consiste em efetuar a melhor distribuição das tarefas e máquina ao longo do tempo. Em se tratando de *Job Shop Problem*, Ahmed *et. al.* (2004) aponta que uma das mais complicadas tarefas é a alocação ótima de recursos, cujo objetivo é sempre a busca de um algoritmo para se efetuar a otimização do uso de tais recursos. Devido à utilização de vários recursos em um ou mais projetos, a elaboração de uma solução para o JSP se torna complexa. Tal complexidade é asseverada por Müller, Rodrigues e Gómez (2006) como sendo um problema de difícil resolução.

Esses problemas são grandes desafios a soluções computacionais e ainda não existe um algoritmo eficiente que garante a obtenção da solução ótima em tempo polinomial (BAKER, 1974). A maioria dos problemas causados pelo JSP é de difícil resolução devido à sua natureza complexa e composta em sua formulação. A complexidade para a solução também é discutida por Reeves (1994), apontando que os problemas de otimização combinatória são problemas onde se busca encontrar um arranjo de variáveis de decisão, representadas por objetos discretos, em que a solução encontrada represente a solução ótima ao problema proposto.

A principal dificuldade em se tratar problemas de otimização combinatória reside no grande número de combinações existentes entre as variáveis de decisão que compõem o modelo do problema. Esse número de combinações resulta num espaço de busca muito grande, impossibilitando a aplicação eficaz de métodos de resolução exatos a um custo computacional razoável (Reeves, 1994). Apesar desta característica, vários pesquisadores estudam métodos de resolução de problemas de otimização combinatória baseada em modelos heurísticos. Heurísticos são métodos que procuram boas soluções, não necessariamente a ótima, existe um custo computacional razoável (Reeves, 1994). As pesquisas nessa área evoluíram os conceitos e implementações destes métodos, de forma que eles tornaram-se mais robustos e versáteis para aplicações a problemas desse tipo.

A complexidade para a solução também é discutida por (Campello e Maculan, 1994; Reeves, 1993), apontando que os problemas de otimização combinatória são problemas onde se busca encontrar um arranjo de variáveis de decisão, representadas por objetos discretos, em que a solução encontrada represente a solução ótima ao problema proposto.

De acordo com Gonçalves, Mendes e Resende (2004), a utilização de modelos matemáticos paramétricos (modelo proposto por Laslo (2010)) para a busca da solução ótima para o JSP tem obtido resultados eficazes para modelos de até 10 x 10 (máquinas x operações). Os modelos de dimensão 15 x 15 (máquinas x operações), os autores apontam que estes são considerados modelos fora do alcance para solução através do uso de algoritmos exatos. Para modelos com estas características ou modelos com dimensões superiores, Gonçalves, Mendes e Resende (2004) asseveram a necessidade do uso de modelos heurísticos (modelos não paramétricos) tais como os modelos propostos por Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006). Um dos principais fatores na formulação do JSP apontado por Solís e Chaves (2004, p. 3) são as restrições relacionadas ao tempo dos Jobs (atividades). As atividades devem respeitar sua data de início, deve respeitar suas relações de precedência e especialmente, a atividade não pode sofrer atraso em sua entrega. Diante dessas restrições, o PMI (2008) indica que o gerente e a equipe do projeto devem perseguir o equilíbrio entre as demandas concorrentes de escopo, prazo, custo, qualidade e riscos. Em relação ao prazo, o modelo proposto por Laslo (2010) faz o tratamento da data de início e término de cada atividade de cada projeto. Em caso de priorização de alguma atividade, será necessária a tomada de decisão pelo gerente de projetos. Para os modelos de Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006), é possível de se efetuar a priorização de atividades. Para isso o modelo dos primeiros autores utiliza como parâmetro de entrada o tempo de duração de cada atividade com o propósito de apresentar como resultado a melhor solução ótima. Já o segundo modelo baseia-se no melhor sequenciamento das atividades para apresentar a solução ótima através da utilização de um algoritmo genético responsável pelo cálculo ótimo do prazo.

O tratamento em relação ao atraso das atividades é abordado de diferentes maneiras entre os modelos deste trabalho. O modelo de Laslo (2010) é o único a calcular as despesas referentes à ociosidade dos recursos humanos especialistas, as despesas do atraso no makesplan e o custo associado à despesa de atraso de cada atividade. O atraso das atividades é aportado por Gonçalves, Mendes e Resende (2004) como parte de um algoritmo responsável pela programação de tempo das atividades, utilizando como parâmetro de entrada o tempo de atraso de cada iteração. O resultado do modelo apresenta uma nova solução ótima, calculando o desvio ocasionado pelo impacto do atraso no makesplan. O modelo proposto por Müller, Rodrigues e Gómez (2006) analisa o atraso de cada par de conjuntos de atividades utilizado no processo de crossover através de um algoritmo genético. O tempo associado ao atraso será

tratado no cálculo da próxima iteração do modelo, desse modo, buscando a melhor solução ótima para o atraso em relação ao makesplan.

O custo é outro fator de grande importância dentro do contexto de JSP. Galway (2004) e Daneshkhah (2004) asseveram que o processo de tomada de decisões para a alocação de recursos é de grande relevância na melhor obtenção de estimativas de custos. Com o objetivo de redução de custos, o planejamento e a alocação de recursos em um ambiente de múltiplos projetos é tratado por Fattahi *et. al.* (2006), Golenko-Ginzburg (2004), Vasconcelos (2007) e Laslo (2010). Os modelos propostos por Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) não tratam nenhum tipo de custos relacionados a desvios ou a penalidades de prazos. O modelo proposto por Laslo (2010) determina o custo para duas situações: (i) custo relacionado à alocação do recurso em uma atividade (job) durante um período de tempo (tempo de ociosidade do recurso) e (ii) o custo pelo atraso na entrega prevista em contrato do projeto (causado pelo desvio de prazo em uma ou mais atividades do projeto).

Devido ao alto nível de complexidade do JSP, caracterizado por Lagewag, Lenstra e Rinnoy Kan (1979) como um problema fatorial e seu nível de complexidade como NP-completo, alguns tipos de modelos matemáticos podem ter comportamentos diferentes em relação ao tratamento do JSP. Pacheco e Santoro (1999) apresentam dois conjuntos de modelos matemáticos para este tratamento: (i) conjuntos de solução ótima (modelos paramétricos e (ii) conjuntos de soluções heurísticas (modelos não paramétricos).

Os modelos de solução ótima, como Laslo (2010), tem a finalidade de determinar o menor tempo para a solução (caminho mais curto) do problema. Assim, Muniz (2009) define que modelos exatos de solução ótima em busca de soluções são considerados computacionalmente impraticáveis para problemas considerados combinatoriais (como o JSP) devido à grande utilização de tempo de processamento para elaborar a solução ótima e por não explorarem de maneira eficaz todas as dimensões de busca do problema. Em contrapartida, modelos heurísticos têm como objetivo apontar a melhor solução, em tempo polinomial, para a solução do JSP. Para a utilização destes modelos de forma computacional, Pacheco e Santoro (1999) definem que modelos que utilizam algoritmos baseados na otimização (minimização) do problema são viáveis apenas quando utilizados para modelos de JSP 10 x 10 devido à dimensão combinatória do problema. Para modelos maiores que 10 x 10 (considerados de grande porte), Pacheco e Santoro (1999) indicam a utilização de modelos matemáticos heurísticos para a solução destes modelos de JSP por obterem tempo de processamento computacional aceitável e por terem ultrapassarem os limites dos métodos de

solução ótima com a finalidade de determinar soluções mais eficazes. Os modelos de propostos por Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) são modelos heurísticos (não paramétricos) baseados em algoritmos genéticos. Algoritmos genéticos são definidos por Pacheco e Santoro (1999) como robustos e seu comportamento é de fácil adaptação e quando são bem estruturados são capazes de resolver problemas de otimização (como o JSP) além de possuírem algumas diferenças em relação aos métodos de busca de solução ótima: (i) trabalham com uma população ao invés de um único ponto, (ii) possuem uma codificação do conjunto de parâmetros de entrada e não somente com os próprios parâmetros e (iii) utilizam regras probabilísticas e não determinísticas.

O Quadro 9 demonstra os modelos propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) e suas respectivas características de uso em relação aos indicadores de avaliação.

Quadro 9 - Modelos utilizados para resolução de JSP.

Requisito	Indicador	Modelos utilizados para resolução de Job Shop Problem		
		Laslo (2010)	Gonçalves, Mendes e Resende (2002)	Müller, Rodrigues e Gómez (2006)
Dimensão	Modelos até 10 x10	Não é especificado a quantidade de modelos utilizado no estudo	Apresentou bons resultados. Resultado de 72% para estes modelos	Para modelos até 10 x 10, o algoritmo proposto atendeu todos os requisitos do estudo em relação a Solução Ótima Conhecida
	Modelos superiores a 10 x10	Não é especificado a quantidade de modelos utilizado no estudo	Nos testes efetuados para este cenário, o modelo de Nowicki and Smutnicki (1996) mostrou maior desempenho para modelos de 15 x 15	Para modelos até 15 x 15, o algoritmo proposto apresentou a maior diferença em relação a Solução Ótima Conhecida 7 pontos (equivalente a 0,57%)
Processamento	Tempo de processamento estimado	Não é especificado	Não é especificado	Não é especificado
	Volume de dados gerados pelas equações	Não é especificado	Utilização de uma lista dos dados para manter o histórico da evolução do processo de busca, pode requisitar mais volume de armazenamento de dados	Não é especificado
Custos	Por Job (atividade)	O modelo trata o custo direto associado ao emprego do recurso por período de tempo e o custo da penalidade por unidade de tempo em atraso.	Não é especificado	Não é especificado
	Total	O modelo trata o custo da penalidade por não conclusão do projeto no prazo contratual	Não é especificado	Não é especificado
Escalonamento	Priorização da data de entrega	Trata o início e o término de cada atividade, sem a possibilidade de priorização.	O modelo faz a priorização das operações (jobs) tendo como parâmetro de entrada o tempo de cada uma das atividades, resultando em apresentar a melhor solução ótima para o problema.	Feito o tratamento para encontrar o melhor sequenciamento das atividades (Jobs) através do uso de um percurso ótimo previamente definido em função do prazo.
	Tratamento ao atraso	Em caso de atraso, o modelo calcula as despesas do Makespan, a ociosidade dos recursos humanos especializados e o custo das despesas de atraso	Tratamento de atraso de cada iteração. O atraso é parte do algoritmo de programação de tempo ativo, tendo impacto direto no makesplan	O atraso entre duas iterações é tratado. Este atraso é automaticamente tratado na iteração seguinte através da função do algoritmo genético.
Características	Modelo matemático usado	Modelo de Solução Ótima (Paramétrico) com utilização de Programação Dinâmica com exercício de Minimização	Modelo Heurístico (Não Paramétrico) com utilização de Algoritmo Genético e distribuição CrossOver	Modelo Heurístico (Não Paramétrico) com utilização de Algoritmo Genético baseado no modelo SMF (Sistema de Manufatura Flexível) proposto por Stecke(1986).
	Uso do modelo em ambiente de softwares	O uso do modelo através de software ou planilha pode se tornar inviável devido a grande quantidade de equações, restrições e parâmetros de entrada dos algoritmos	Construção através do uso de software especializado, respeitando as regras do modelo.	Construção através do uso de software especializado, respeitando as regras do modelo.
Resumo	Características apropriadas de uso	<ul style="list-style-type: none"> - Modelos JSP até 10 x 10 - Necessidade de informações de custos associados ao desvio de prazo das atividades - Necessidade de informações de custos associados a ociosidade dos recursos humanos especialistas - Cálculo das despesas em caso de atraso no prazo de entrega contratual - Pouco espaço de armazenamento computacional - Tratamento ao atraso leva-se em conta a ociosidade dos RHE 	<ul style="list-style-type: none"> - Modelos JSP superiores a 10 x 10 - Maior flexibilidade de construção do modelo devido a utilização de algoritmo genético - Não há necessidade de informações referentes a custos de desvio de prazo e ociosidade de recursos humanos especialistas - Priorização de atividades utilizando como regra o melhor sequenciamento entre as atividades - Tratamento do atraso da iteração atual com impacto no makesplan - Uso de base histórica pelo algoritmo - Construção do modelo através de desenvolvimento de software específico 	<ul style="list-style-type: none"> - Modelos JSP superiores a 10 x 10 - Maior flexibilidade de construção do modelo devido a utilização de algoritmo genético - Não há necessidade de informações referentes a custos de desvio de prazo e ociosidade de recursos humanos especialistas - Priorização de atividades tendo como base a função em relação ao prazo total - Pouco espaço de armazenamento computacional - Tratamento do atraso da iteração atual calculado na iteração seguinte - Construção do modelo através de desenvolvimento de software específico

Fonte: Elaborado pelo autor.

4.5 REQUISITOS DE ADERÊNCIA DOS MODELOS X SOLUÇÃO JSP

O Quadro 10 apresenta as contribuições dos modelos propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004), e seus respectivos resultados para o tratamento de JSP, por meio da avaliação de indicadores pré-definidos.

Quadro 10 - Indicadores de eficácia do modelo de otimização propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) para a solução de JSP.

Requisito	Indicador	Resultado
Dimensão	Modelos até 10x10 (quantidade de jobs x quantidade recursos)	De acordo com Gonçalves, Mendes e Resende (2004), a utilização de modelos matemáticos paramétricos - modelo proposto por Laslo (2010) - tem obtido resultados eficazes em busca da solução ótima para modelos de JSP de até 10x10.
	Modelos superiores a 10x10 (quantidade de máquinas x quantidade de jobs - operações)	Para modelos com dimensões superiores a 10x10, Gonçalves, Mendes e Resende (2004) e Müller e Rodrigues e Gómez (2006) asseveram a necessidade do uso de modelos heurísticos (modelos Não Paramétricos).
Processamento	Tempo de processamento estimado	Indicador não é tratado em nenhum dos três modelos propostos neste estudo.
	Volume de dados gerados pelas equações	Tratado apenas pelo modelo proposto por Gonçalves, Mendes e Resende (2004), onde o algoritmo genético responsável pela elaboração do caminho ótimo das atividades.
Custos	Por job (atividade)	O modelo de Laslo (2010) trata o custo direto associado ao emprego do recurso por período de tempo e o custo da penalidade por unidade de tempo em atraso. Os modelos de Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) não abordam este requisito.
	Total	O modelo de Laslo (2010) trata o custo da penalidade por não conclusão do projeto no prazo contratual. Os modelos de Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) não abordam este requisito.
Escalonamento	Priorização da data	O modelo de Laslo (2010) trata o início e o término de cada atividade, sem a possibilidade de

	de entrega	<p>priorização.</p> <p>O modelo de Gonçalves, Mendes e Resende (2004) permite a priorização levando em consideração o tempo de cada atividade.</p> <p>Já o modelo de e Müller, Rodrigues e Gómez (2006) procura efetuar o melhor sequenciamento das atividades através do seu percurso ótimo, sem a possibilidade de priorização.</p>
	Tratamento ao atraso	<p>Em caso de atraso, o modelo de Laslo (2010) calcula as despesas do makesplan, a ociosidade dos recursos humanos especialistas e o custo das despesas de atraso.</p> <p>Gonçalves, Mendes e Resende (2004) em seu modelo abordam o indicador efetuando o tratamento de cada iteração através de seu algoritmo de programação.</p> <p>O modelo proposto por Müller, Rodrigues e Gómez (2006) efetua o tratamento ao atraso de cada iteração através de algoritmo genético.</p>
Características	Modelo matemático usado	<p>O modelo de Laslo (2010) é um modelo matemático de solução ótima (Paramétrico) com utilização de Programação Dinâmica com exercício de Minimização.</p> <p>Os modelos de Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) são modelos matemáticos heurísticos (não paramétricos) baseados em algoritmos genéticos. O modelo de Gonçalves, Mendes e Resende (2004) utiliza-se de distribuição CrossOver para a solução do JSP, enquanto que o modelo de Müller, Rodrigues e Gómez (2006) faz o tratamento do JSP levando-se em consideração o algoritmo genético SMF proposto por Stecker (1986).</p>
	Uso do modelo em ambiente de softwares	<p>O uso do modelo de Laslo (2010) através de software ou planilha pode se tornar inviável devido a grande quantidade de equações, restrições e parâmetros de entrada dos algoritmos.</p> <p>A construção de um software especializado ou planilha eletrônica, respeitando as regras de confecção do modelo é viável para os modelos propostos por modelos de Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006).</p>

Fonte: Elaborado pelo autor.

4.6 AVALIAÇÃO DOS MODELOS EM ESTUDO DE CASO

4.6.1 CARACTERIZAÇÃO DA EMPRESA

A empresa abordada nesta pesquisa é uma companhia líder no mercado de idioma hispano-português de desenvolvimento de softwares para Contact Center. Iniciou suas atividades no Brasil no estado de São Paulo, no ano de 1999. A partir de 2000 começou a expansão de seus negócios com a criação de novos serviços nas áreas de finanças, telecomunicações, seguros, tecnologia e indústrias, alcançando como resultado o status de uma empresa de Contact Center com a prestação de serviços em múltiplas plataformas e tecnologias.

Atualmente a empresa é considerada uma das maiores empregadoras privadas no território brasileiro além de ser uma das mais importantes formadoras de mão-de-obra qualificada no Brasil.

Alguns dados importantes da atuação da empresa no Brasil no ano de 2010 são descritos a seguir:

- a) atuação em 17 países;
- b) em 2010 a receita da empresa apresentou um crescimento de 1.663 bilhões de euros;
- c) desde o ano 2000, a empresa multiplicou por três o número de funcionários, passando de 40.324 para 152.000;
- d) hoje é um dos principais empregadores nos países em que atua;
- e) administra operações em todos os níveis de complexidade e tamanho;
- f) possui mais de 500 clientes de diversos setores em todo o mundo;
- g) é a terceira maior empresa de Contact Center do mundo;
- h) é líder nos países de idioma hispano-português.

Os ramos de negócios da empresa objeto deste trabalho estão divididos em três regiões ao redor do mundo: (i) Brasil (país onde a empresa possui elevado volume de receita); (ii) Américas (nesta região compreende as operações do continente Americano, com exceção do Brasil, (EUA (Texas e Porto Rico), Argentina, Chile, Colômbia, El Salvador, Guatemala,

México, Panamá, Peru, Uruguai e Venezuela) e (iii) EMEAA –Europa Oriente Médio, África e Ásia (região que compreende Espanha, França, Marrocos e República Checa).

4.6.2 CENÁRIO ATUAL DA EMPRESA

A partir do início das atividades da empresa no Brasil, rapidamente o número de projetos de desenvolvimento de software cresceu de forma abrupta devido ao crescimento de sua carteira de clientes. A rigor, a área responsável pelo desenvolvimento de softwares cresceu no ritmo da empresa. Com o aumento de projetos e a escassez de recursos humanos especialistas dentro da área de desenvolvimento de softwares da empresa, a disputa por esse tipo de recurso humano se tornou acirrada.

Conforme apontado por Boff e Abel (2005), este tipo de recurso é caracterizado pelo fato de executar atividades de alta complexidade através da utilização de recursos tecnológicos em suas atividades (Jobs). Outra grande característica dos recursos humanos especialistas dentro de um ambiente de desenvolvimento de softwares é apontada por Boff e Abel (2005). Os autores citam que estes profissionais são exigidos em projetos devido ao seu alto nível de habilidade e o alto grau de experiência, o que pode aumentar ainda mais a concorrência dentro de uma mesma empresa.

O vínculo de dependência entre a área de desenvolvimento de projetos de softwares e a necessidade de utilização de recursos humanos especialistas é assinalado por Boff e Abel (2005), que determinam que a dependência em relação a esse tipo de profissional é caracterizada por este recurso ser um indivíduo que tem sua alocação em uma determinada atividade (job) estratégica da empresa de maneira intensificada. Os resultados desta relação são asseverados por Boff e Abel (2005) que determinam que os recursos humanos especialistas possuam um índice de 85% de aproveitamento em resolução de atividades críticas contra 30% para os demais recursos.

4.6.3 PROBLEMAS ENFRENTADOS PELA EMPRESA

A disputa acirrada pela alocação de recursos humanos especialistas em projetos de desenvolvimento de software cresceu de forma abrupta devido ao crescimento de sua carteira

de clientes. Em se tratando desenvolvimento de software, Fairley (1994) estabelece que muitos destes projetos não são entregues dentro do prazo acordado e dentro do custo planejado. Os impactos estão diretamente associados ao ambiente interno das empresas, conforme apontado por Campanario *et. al.* (2009), diferentes projetos concorrem com uma quantidade limitada de recursos humanos.

Tal fato é enfatizado por Pressman (2006), apontando que problemas relacionados ao escalonamento e sequenciamento de atividades podem ocorrer dentro de um ambiente caracterizado pela produção sob encomenda, foco no desenvolvimento e ênfase no ciclo de vida do produto como o ambiente de desenvolvimento de software. Um dos objetivos do sequenciamento de atividades, de acordo com Muniz (2009) é minimizar ou maximizar o prazo ou o custo do desenvolvimento. Baker (1974) aponta que se o processo de sequenciamento e escalonamento de atividades não for efetuado de forma correta, problemas relacionados à alocação de recursos humanos especialistas ao longo do tempo podem ocorrer.

Para demonstrar os impactos causados pelos problemas de sequenciamento de atividades, escalonamento de recursos humanos especialistas ao longo do tempo e por problemas de desvio de prazos de projetos de desenvolvimento de software dentro da empresa origem deste estudo, o Quadro 11 apresenta a quantidade anual de projetos de desenvolvimento de software e a quantidade de projetos com desvio de prazo (causados por problemas de escalonamento de recursos e sequenciamento de atividades), durante o período de Janeiro 2008 até Junho de 2010.

Quadro 11 - Quantidade total de projetos e projetos com problemas de desvio de prazos originados por sequenciamento de atividades e escalonamento de recurso.

Ano	Total de projetos	Projetos com desvio de prazo
2008	150	42
2009	176	48
2010	96	17

Fonte: Elaborado pelo autor.

Em suma, os problemas dos desvios de prazos dos projetos de desenvolvimento de software se processaram em meio a:

- a) escassez de recursos humanos especialistas;
- b) competitividade por utilização de recursos humanos especialistas nos projetos;
- c) complexidade em se calcular os custos ocasionados pelos desvios (custo total e por desvio de atividade);
- d) dificuldades em efetuar a priorização de projetos com o objetivo de minimizar os problemas nos atrasos dos projetos.

4.6.4 AVALIAÇÃO DOS MODELOS PARA RESOLUÇÃO DO JSP PARA O ESTUDO DE CASO

Para a validação dos resultados obtidos por meio da avaliação dos requisitos descritos anteriormente, foi avaliada a aderência do uso dos modelos propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) na prática, em uma empresa de desenvolvimento de software. Foi realizada uma entrevista estruturada com o gestor de projetos da empresa analisada por esta pesquisa, de acordo com a seguinte estrutura, demonstrada no quadro 12. O gestor de projetos é responsável por toda a carteira de projetos financeiros da empresa analisada, participando na tomada de decisões desde a fase de planejamento até o acompanhamento pós-implantação de todos os projetos.

Quadro 12 - Aspectos relacionados a entrevista realizada com o gestor de projetos de uma empresa desenvolvimento de software.

Pergunta 1:	Qual a média anual da alocação de recursos (humanos e máquinas) nos projetos de sua competência?
Resposta:	11 recursos, compostos por analistas de sistemas e programadores.
Pergunta 2:	Como são tratados os atrasos em relação aos projetos?
Resposta:	Todo projeto com risco de atraso passa por uma análise onde são verificados os motivos do atraso, seu impacto e as alternativas possíveis para recuperá-lo. A decisão de que ação será tomada varia de acordo com o projeto, pois depende do impacto que esse atraso trará ao negócio e da viabilidade da ação. Normalmente são 3 opções discutidas:

	<p>a) negociação com cliente para prorrogação do prazo.</p> <p>b) super-alocação de recursos (uso de hora extra).</p> <p>c) alocação de recursos adicionais.</p>
Pergunta 3:	Em caso de desvio de prazo de entrega do projeto, quais são os fatores críticos que são analisados?
Resposta:	<p>São analisados os motivos do atraso, que normalmente podem ser (i) o atraso originado no cliente, (ii) a desempenho/produktividade abaixo da expectativa, (iii) a alteração de escopo não controlado, (iv) erro nas estimativas das atividades, (v) o atraso por parte do fornecedor e (vi) a ausências/faltas não planejadas de recursos.</p> <p>Também são analisados os impactos no negócio, como (i) a relação direta com processos de negócio ex: regra obrigatória a partir de determinada data, (ii) perdas financeiras decorrentes do atraso, (iii) os custos associados, (iv) impacto na imagem da empresa e (v) impacto na satisfação do cliente.</p> <p>A partir dessas informações serão analisadas as alternativas para recuperação ou adiamento do projeto.</p>
Pergunta 4:	Como é feita a alocação de recursos humanos especialistas nos projetos, principalmente para projetos com alto nível de criticidade?
Resposta:	<p>São avaliados os seguintes itens para alocação do recurso:</p> <p>a) complexidade/característica da demanda x perfil dos recursos da equipe.</p> <p>b) urgência/prazo da entrega.</p> <p>c) disponibilidade dos recursos.</p> <p>d) projetos em andamento e na fila de espera.</p>
Pergunta 5:	A priorização da data de entrega dos projetos é uma prática frequente dentro da carteira de projetos?
Resposta:	Sim. Muitas vezes os projetos estão associados a um início de vigência de regras de negócio, início de vigência de legislação ou necessidades específicas de negócio que demandam re-priorização, revisão das alocações e reprogramação de datas de entrega dos projetos.
Pergunta 6:	Como um modelo matemático poderia ajudar no processo de alocação de recursos em projetos?
Resposta:	<p>Um modelo matemático poderá ajudar contemplando os seguintes requisitos:</p> <p>a) servir como ferramenta de apoio no processo de alocações de forma a agilizar e facilitar a identificação de melhor cenário de alocação e;</p> <p>b) redução de ociosidade e consequentemente de custos por alocação mais</p>

	adequada dos recursos.
--	------------------------

Fonte: Elaborado pelo autor.

Com base nas respostas descritas no Quadro 12, o Quadro 13 demonstra a aderência dos modelos de otimização proposto por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) a cada indicador de eficácia ao ambiente de desenvolvimento de software da empresa objeto deste estudo:

Quadro 13 - Aderência dos modelos de Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) x indicador de eficácia.

Requisito	Indicador	Laslo (2010)	Gonçalves, Mendes e Resende (2004)	Müller, Rodrigues e Gómez (2006)
Dimensão	Modelos até 10x10	Aderente	Não Aderente	Não Aderente
	Modelos superiores a 10x10	Não Aderente	Aderente	Aderente
Processamento	Tempo de processamento estimado	Não Aderente	Não Aderente	Não Aderente
	Volume de dados gerados pelas equações	Não Aderente	Não Aderente	Aderente
Custos	Por Job (Atividade)	Aderente	Não Aderente	Não Aderente
	Total	Aderente	Não Aderente	Não Aderente
Escalonamento	Priorização da data de entrega	Não Aderente	Aderente	Não Aderente
	Tratamento ao atraso	Aderente	Aderente	Aderente
Características	Modelo matemático usado	Não Aderente	Aderente	Aderente
	Uso do modelo em ambiente de softwares	Não Aderente	Aderente	Aderente

Fonte: Elaborado pelo autor.

A partir dos resultados obtidos anteriormente é possível fazer as seguintes considerações:

a) Em relação à dimensão do modelo de JSP:

Os modelos propostos por Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) são eficazes para modelos de JSP superiores a 10x10 (quantidade de jobs x quantidade recursos) – modelo de JSP encontrado na empresa objeto deste estudo, tornado o indicador “dimensão” do modelo aderente ao ambiente da empresa.

Já o modelo proposto por Laslo (2010) não é aderente ao indicador “dimensão”, uma vez que este modelo é eficaz somente para modelos de JSP até 10x10 - modelo não utilizado pela empresa.

b) Em relação ao Processamento:

Nenhum dos três modelos utilizados neste estudo abordou o indicador “Tempo de processamento estimado”, tornando-o não aderente.

O indicador “Volume de dados gerados pelas equações” é tratado apenas pelo modelo proposto por Gonçalves, Mendes e Resende (2004). O algoritmo genético deste modelo utiliza uma base histórica de informações para cálculo da próxima iteração de atividades. A dimensão do modelo do JSP influencia diretamente no volume de dados a serem armazenado na base histórica. Quanto maior a dimensão do modelo de JSP maior será o espaço para armazenamento da base histórica.

c) Em relação aos custos:

A empresa necessita do custo por atividade devido à restrição e ociosidade de recursos humanos e recursos humanos especialistas disponíveis. A empresa também necessita do custo por penalidade por desvio de prazo total para a entrega do projeto. O modelo de Laslo (2010) efetua o cálculo das despesas referentes à ociosidade dos recursos humanos especialistas, as despesas do atraso no makesplan e o custo associado à despesa de atraso de cada atividade, tornando os indicadores de custo (“Custo por job” e “Custo Total”) deste modelo aderente ao ambiente da empresa.

Os modelos propostos por Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) não abordam os indicadores de custos.

d) Em relação ao escalonamento:

Devido à grande volatilidade de alocação dos recursos humanos e recursos humanos especialistas e do nível de criticidade dos projetos, a priorização da data de entrega dos processos é uma prática utilizada pela empresa.

A priorização da data de entrega é aderente apenas pelo modelo proposto por Gonçalves, Mendes e Resende (2004). Modelo este que se baseia no tempo de todas as atividades para recalcular o caminho ótimo respeitando a priorização solicitada.

O modelo proposto por Laslo (2010) e por Müller, Rodrigues e Gómez (2006) não permitem tratamento ao indicador “Priorização da data de entrega”.

Para a empresa, a informação do custo de ociosidade dos recursos humanos especialistas é de grande importância e atualmente o gestor dos projetos não possui esta informação. Desse modo, o indicador “Tratamento ao atraso” se torna aderente ao ambiente da empresa para os três modelos apresentados neste estudo. O que varia entre os modelos é como o cálculo da ociosidade é efetuado. Laslo (2010) calcula as despesas do makesplan, a ociosidade dos recursos humanos especialistas e o custo das despesas de atraso. Gonçalves, Mendes e Resende (2004) efetuam o cálculo a partir de seu algoritmo genético de programação do sequenciamento das atividades e, por fim, o cálculo da ociosidade é tratado por Müller, Rodrigues e Gómez (2006) através do algoritmo genético em cada ponto de atraso entre as iterações das atividades.

e) Em relação às características do modelo:

Os modelos de solução ótima, como Laslo (2010), tem a finalidade de determinar o menor tempo para a solução (caminho mais curto) do problema. Para a utilização destes modelos de forma computacional, Pacheco e Santoro (1999) definem que modelos que utilizam algoritmos baseados na otimização (minimização) do problema são viáveis apenas quando utilizados para modelos de JSP 10 x 10 devido à dimensão combinatória do problema, tornando os indicadores do requisito “Características” não aderente à necessidade da empresa para o modelo proposto por Laslo (2010).

Em relação às características da empresa em quanto à dimensão do modelo de JSP (a empresa objeto deste estudo possui modelo de JSP maior que 10x10), Pacheco e Santoro (1999) indicam a utilização de modelos matemáticos heurísticos, como os modelos propostos por Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) para a solução destes modelos de JSP. Os algoritmos utilizados por estes modelos possuem menor tempo de processamento computacional na busca de se ficar a melhor solução ótima de maneira mais eficaz, tornando os indicadores “Modelo matemático usado” e “Uso do modelo em ambiente de softwares” aderente aos modelos propostos por Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004).

4.6.5 DISCUSSÃO

O modelo proposto por Laslo (2010), por se tratar de ser um modelo matemático paramétrico, possui comportamento eficaz em modelos de JSP com dimensão até 10x10 (10 máquinas x 10 atividades – Jobs), enquanto que os modelos de Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) são caracterizados por serem modelos matemáticos não paramétricos, sendo assim, mais eficazes quando utilizados em modelos de JSP com dimensão superior a 10x10 (10 máquinas x 10 atividades – Jobs).

Em relação aos custos referentes a atrasos, o modelo de Laslo (2010) se apresenta de maneira expressiva efetuando o cálculo do custo por atividade (custo associado ao recurso e custo associado às penas contratuais da atividade) e do custo total de cada projeto, enquanto que os modelos Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) não fazem tratamentos aos custos, tanto os custos totais ou por atividades, causados por desvios de prazos.

A priorização de atividades é abordada apenas pelo modelo proposto Gonçalves, Mendes e Resende (2004). O Modelo de Laslo (2010) apenas efetua o tratamento dos atrasos de cada atividade. Já o modelo proposto por Müller, Rodrigues e Gómez (2006) realiza o melhor sequenciamento das atividades levando em consideração a estimativa de duração de cada atividade, sem possibilidade de priorização.

Em se tratando de atraso das atividades, o modelo de Laslo (2010) faz o tratamento dos custos causados pela ociosidade dos recursos humanos e principalmente, recursos humanos especialistas. Os modelos de Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) tratam apenas os atrasos relacionados aos desvios de prazos das atividades, não tratando a ociosidade dos recursos disponíveis à alocação.

O uso do modelo de Laslo (2010) através de um software customizado ou por meio de planilhas pode ser uma tarefa árdua devido a grande quantidade de equações, restrições e parâmetros de entrada necessários para a execução do modelo. De forma análoga, os modelos proposto por Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) são modelos elaborados para utilização de algoritmos genéticos, possuindo assim característica de maior adaptação, vigor na resolução de problemas de otimização como o JSP. Com estas características, se torna possível a construção de um software customizado para a aplicação dos modelos de Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004).

A partir de uma entrevista com o gestor de projetos de uma empresa de desenvolvimento de softwares foi possível identificar através das respostas, a aderência dos indicadores à necessidade de resolver os problemas de escalonamento e alocação de recursos causados pelo JSP. Ficou evidente que apenas os modelos propostos por Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) são eficazes em modelos de JSP superiores a 10x10, modelo de JSP que reflete o cenário atual da empresa.

Em relação aos indicadores de custos, foi identificado que uma das maiores dificuldades da empresa é calcular o custo associado à ociosidade dos recursos humanos nos projetos. Este indicador somente é tratado pelo modelo proposto por Laslo (2010), que também é o único modelo que aborda o custo associado ao atraso de atividades.

Já para as características dos modelos apresentados neste estudo, os indicadores “Modelo matemático usado” e “Uso do modelo em ambiente de software” estão em conformidade somente para os modelos propostos por Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006), pelo fato que o modelo de JSP apresentado pela empresa (superior a 10x10) é aderente aos modelos em busca da melhor solução para o problema de alocação de recursos, além de ser possível a construção dos modelos com a utilização de algum software ou planilha especializada. O fato do modelo de JSP apresentado pela empresa neste estudo ser superior a 10x10 e a complexidades das equações matemáticas utilizadas pelo modelo, tornam ineficiente o uso do modelo proposto por Laslo (2010) na busca da melhor solução ótima para o problema de otimização de recursos.

Para o requisito de escalonamento, o indicador “Tratamento ao atraso” é aderente aos três modelos utilizados neste estudo, variando apenas a forma como são abordados por seus algoritmos internos para o cálculo do sequenciamento das atividades. Em relação ao indicador “Priorização da data de entrega”, apenas o modelo proposto por Gonçalves, Mendes e Resende (2004) é aderente, uma vez que a priorização da data de entrega dos processos é uma prática utilizada pela empresa em seus projetos.

Pelo fato de a empresa possuir um cenário onde fica incontestável a existência de modelo de JSP superior a 10x10, para se atingir uma melhor eficácia na busca pela melhor solução ótima, torna-se necessário a utilização de modelos matemáticos heurísticos (não paramétricos).

A empresa efetua, em seus projetos, o cálculo dos custos (total e por atividade) causados por desvios de prazo. Como os modelos propostos por Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) não abordam estes indicadores. Neste caso, recomenda-se a utilização do modelo de Laslo (2010). Ainda em se tratando de custos, a empresa possui a necessidade de se calcular o custo da ociosidade de recursos, principalmente recursos humanos especialistas. Para este fator, é recomendada também a utilização do modelo de Laslo (2010), pois somente este modelo trata este requisito.

Outro indicador contido no ambiente de desenvolvimento de softwares da empresa é a priorização de atividades. Esta prática ocorre de maneira constante dentro do ambiente de projetos da empresa. Para este indicador, é aconselhado o uso modelo proposto por Gonçalves, Mendes e Resende (2004).

5 CONCLUSÃO

O presente trabalho procurou avaliar as contribuições dos modelos de otimização propostos por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) para a resolução prática de problemas de JSP em projetos em um ambiente de desenvolvimento de software. Como discutido, um dos grandes desafios destas empresas é quando diferentes projetos necessitam ou concorrem por uma quantidade limitada de um mesmo recurso humano. Este cenário torna-se agravante quando a disputa é em relação a recursos que executam tarefas de alta criticidade, denominados recursos humanos especialistas. Este cenário contribui para a formulação de um dos maiores desafios a serem solucionados – o JSP. As informações apresentadas neste estudo têm como objetivo analisar os modelos matemáticos para o tratamento de JSP proposto por Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) em projetos de desenvolvimento de software. Para isso, foram observadas algumas características desses modelos em relação ao tratamento do JSP (Quadro 14).

Quadro 14 - Características dos modelos de Laslo (2010), Müller, Rodrigues e Gómez (2006) e Gonçalves, Mendes e Resende (2004) em relação ao tratamento do JSP.

Modelo Matemático	Características
Laslo (2010)	<ul style="list-style-type: none"> • Modelo matemático paramétrico • Baseado em otimização de recursos • Minimização de penas contratuais • Transferência de recursos em diversos projetos por um período pré-determinado
Gonçalves, Mendes e Resende (2004)	<ul style="list-style-type: none"> • Modelo matemático não paramétrico • Chaves aleatórias para o tratamento do tempo • Algoritmo genético para definição de prioridades e parametrização de tempos das atividades • Cronograma com a melhor solução
Müller, Rodrigues e Gómez (2006)	<ul style="list-style-type: none"> • Modelo matemático não paramétrico • Utiliza o tempo total de parada e atraso das atividades • Algoritmo genético responsável pelo sequenciamento das atividades • Busca do melhor sequenciamento de atividades

Fonte: Elaborado pelo autor.

Cada modelo abordado neste estudo possui suas características próprias em relação ao tratamento ao JSP, devido ao modelo matemático utilizado e principalmente em relação ao tipo de algoritmo usado para a resolução do problema de otimização de recursos. Para melhor aproveitamento de cada modelo dentro de um ambiente de desenvolvimento de software, é necessário identificar os aspectos da empresa, como a necessidade de se calcular os riscos em relação aos atrasos, a prática de priorização das atividades e principalmente, o tamanho do modelo de JSP encontrado dentro do ambiente de desenvolvimento de softwares da empresa. De posse das informações apontadas anteriormente, é possível traçar a aderência das necessidades da empresa em relação ao modelo matemático a ser utilizado para o tratamento do JSP.

Esta pesquisa teve como objetivo avaliar os modelos matemáticos de algoritmo genético proposto por Laslo (2010) e de otimização propostos por Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) e suas respectivas contribuições para resolução de *Job Shop Problem* em projetos de desenvolvimento de software. Este trabalho não teve o propósito de eleger o melhor modelo para resolução de JSP. Para essa conclusão, é interessante a realização de pesquisas com uso de simulações numéricas a fim de comparar os resultados gerados para cada modelo, uma vez que Pacheco e Santoro (1999) definem que um modelo matemático mais adequado para tratamento do JSP seria baseado na união dos modelos exato e heurístico, juntando as melhores técnicas entre ambos na busca da melhor solução ótima.

Como sugestão para trabalhos futuros destaca-se a realização de pesquisas relacionando a aplicação dos modelos propostos por Laslo (2010), Gonçalves, Mendes e Resende (2004) e Müller, Rodrigues e Gómez (2006) em relação à aderência de outros indicadores. Entende-se também como relevante uma análise comparativa através da utilização de outros modelos de otimização para a resolução do problema de JSP em projetos.

REFERÊNCIAS

- ACEVEDO, C. R.; NOHARA, J. J. **Monografia no curso de administração, guia completo de conteúdo e forma**. 2.a edição, Editora Atlas, São Paulo, 2006.
- ABEL, M. **Estudo da perícia em petrografia sedimentar e sua importância para a engenharia do conhecimento**. 2001. 239 f. Tese (Doutorado) - Programa de Pós Graduação em Ciência da Computação, Porto Alegre, 2001.
- AHMED, P.; MOGHADDAM, R. T.; JOLAI, F; VAZIRI, F. Solving stochastic job shop scheduling problems by a hybrid method. **University of Wolverhampton**, 2004.
- ANDERSEN, E. S.; JESSEN, S. A. Project maturity in organizations. **International Journal of Project Management**, vol. 21, p. 457-461, 2002.
- ARCHER, N., GHASEMZADEH, F., A decision support system for project portfolio selection. **International Journal of Technology Management**, Vol. 16, No 1-3, p.105-114, 1998.
- ARTTO, K., MARTINSUO, M., AALTO, T., Project portfolio management: strategic management through projects. **Project Management Association Finland**, p. 5-22, 2001.
- BAKER, K. R. **Introduction to sequencing and scheduling**. Canada: John Wiley and SonsInc, 1974.
- BALLARD, D., BROWN, C. **Computer vision**, New Jersey: Prentice-Hall, 1982.
- BOEHM, B. A spiral model of software development and enhancement. **IEEE Software**, vol. 21, p. 61-72, 1988.
- BOEHM, B. Software risk management: principles and practices. **IEEE Software**, vol. 8, p. 32-41, 1990.
- BOEHM, B.; DEMARCO, T. Software risk management. **IEEE Software**, p. 17-19, 1997.
- BOFF, L. H.; ABEL, M. Autodesenvolvimento e Competências: o caso do trabalhador de conhecimento como especialista. In RUAS, R. et alii. **Aprendizagem Organizacional e Competências**. Porto Alegre: Bookman, 2005.
- CALLEGARI-JACQUES, S. M. **Bioestatística: Princípios e Aplicações**. Porto Alegre: Artmed, 2003.
- CAMPANARIO, M. A.; MACCARI, E. A.; SILVA, M. M.; SANTANA, S. G. Desenvolvimento de um curso de mestrado profissional sob a perspectiva de gestão de projetos. **Revista Brasileira de Gestão de Negócios (São Paulo. Impresso)**, v. 11, p. 423-442, 2009.
- CARVALHO, M. M.; RABECHINI JR, R. **Construindo competências para gerenciar projetos**. São Paulo: Atlas, 2009.

CASSARRO, A. C. **Sistemas de informações para tomada de decisões**. São Paulo: Pioneira, 1988.

CAUTELA, A. L.; POLLONI, E. G. F. **Sistemas de informação**. Rio de Janeiro: McGraw-Hill, 1982.

CHAGAS, M., CABRAL, A. Criação de capacitações em integração de sistemas: o caso do programa CBERS DOI: 10.5585/rai.v7i2.431. **Revista de Administração e Inovação**, América do Norte, 7, ago. 2010. Disponível em: <http://www.revistarai.org/ojs-2.2.4/index.php/rai/article/view/431/290>. Acesso em: 14/08/2011.

CHAGAS JUNIOR, M. F.; DUARTE, C. C. M. Arquitetura de Sistemas, conhecimento processual e a exploração de economias de repetição e recombinação. In: **XIV Congresso Latino-Iberoamericano de Gestion Tecnológica**, Peru, 2011.

CHAPMAN, C.; WARD, S. **Project risk management: processes, techniques, and insights**. Chichester: John Wiley & Sons, 1997.

CHURCHMAN, C. West. **Introdução à teoria dos sistemas**. Petrópolis: Vozes, 1972.

CLARK, K. B.; WHEELWRIGHT, S. C. **Managing new product and process development**. New York: The Free Press, 1993.

CLARKE, A. A practical use of key success factors to improve the effectiveness of project management. **International Journal of Project Management**, v. 7, n.3, p. 139-145, 1999.

CLELAND, D. I.; IRELAND, L. R. **Project manager's portable handbook**. 2. ed. New York: McGraw Hill, 2004.

COLLINS, R. W. **Impact of information technology on the process and performance of knowledge workers**. University of Minnesota: Minnesota, 1993.

COOPER; D., R.; SCHINDLER, P. S. **Métodos de pesquisa em administração**. Porto Alegre: Bookman, 2008.

COOPER, R.; EDGETT, S. J.; KLEINSCHMIDT, E. J. New problems, new solutions: making portfolio management more effective. **Research Technology Management**, v.43. n. 2, p.18-33, 2000.

COOPER, R. **Winning at new products – accelerating the process from idea to launch**. Cambridge: Perseus Books, 1993.

CORTADA, J. W. Where did knowledge workers come from? **Rise of the knowledge worker**, p. 3-21, 1998.

DANESHKHAH, A. R. Psychological aspects influencing elicitation of subjective probability. **The University Of Sheffield**. August, 2004.

DAVIS, G.; COLLINS, R. W.; EIERMAN, M.; NANCE, W. D. Conceptual model of research on knowledge work. **Management Information Systems Research Center (MISRC)**, 1991.

DEMING, E. W. **A nova economia para a indústria, o governo e a educação**. Rio de Janeiro: Qualitymark, 1997.

DEY, P.K.; KINCH, J.; OGUNLANA, S.O. Managing risk in software development projects: a case study. **Industrial Management and Data Systems**, p. 284-303, 2007.

DRUCKER, P.F. Knowledge-worker productivity. **California Management Review**, v. 41, n° 2, p.79-94, 1999.

ERICSSON, K. A.; SMITH, J. **Toward a general theory of expertise: prospects and limits**. Cambridge University Press: Cambridge , 1991.

FAIRLEY, R. Risk management for software's projects. **IEEE Software**. p 54-66, 1994.

FATTAHI, P.; MEHRABAD, M. S.; ARYANEZHAD, M.B. An algorithm for multi-objective job shop scheduling problem. **Journal of Industrial Engineering International**, Vol. 2006, N. 3, p. 43-53, 2006.

FERREIRA, O. J. **Algoritmos de programação dinâmica usados em modelos markovianos ocultos (hmms)**. 2005. 250 f. Tese (Doutorado) – Instituto Nacional de Pesquisa Espacial, INPE, São Paulo, 2005.

GALWAY, L. A. **Risk Analysis for Complex Projects**, RAND Corporation (WR-112), 2004.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4.a edição, Editora Atlas, São Paulo, 2006.

GILB, T. **Principles of software engineering management**, Addison-Wesley, 1990.

GOLENKO-GINZBURG, D.; LASLO, Z. Chance constrained oriented dispatching rules for flexible job shop scheduling. **Computer Modelling & New Technologies**, Vol.8, N. 2, p. 14-18, 2004.

GONÇALVES, J.F.; BEIRÃO, N.C. Um algoritmo genético baseado em chaves aleatórias para sequenciamento de operações. **Revista Associação Portuguesa de Desenvolvimento e Investigação Operacional**, Vol. 19, p. 123-137, 1999.

GONÇALVES, J.F.; MENDES, J.M.; RESENDE M.C.G. A hybrid genetic algorithm for the job shop scheduling problem. **European Journal of Operational Research**. To appear, 2004.

GUIMARÃES, K. F. **Escalonamento genético FJSP com tempo de configuração dependente da sequencia**. 2007. 118 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Uberlândia, Minas Gerais, 2007.

GUSMÃO, C.; DE MOURA, H. P. Gerência de risco em processos de qualidade de software: uma análise comparativa. **Anais do III Simpósio Brasileiro de Qualidade de Software**. Brasília – DF – Brasil, 2002.

JUN-GUANG Z.; ZHEN-CHAO, X. Method study of software project risk management. **International conference on computer application and system modeling**. p. 8-11, 2010.

KAPLAN, R. S.; NORTON, D. P. **A estratégia em ação: balanced scorecard**. 2 ed. Rio de Janeiro: Campus, 1997.

KWAK, Y. H.; STODDARD, J. Project risk management: lessons learned from software development environment. **Technovation**, Vol. 24, No.11, p. 915-920, 2004.

KERZNER, H. **Project management: A systems approach to planning, scheduling and controlling**. Tenth Edition. ed. New York: John Wiley and Sons, 2009.

KOONTZ, H.; O'DONNELL, C. **Os princípios de administração: Uma análise das funções administrativas**. São Paulo: Pioneira, 1980.

KUSIAK, A.; CHOW, W. Efficient solving of the group technology problem. **Journal of Manufacturing**, p. 117-124, 1987.

LAGEWAG, B. J.; LENSTRA, J. K.; RINNOOY KAN, A. H. G. Job shop scheduling by implicit enumeration. **Management Science**, vol. 24, p. 441-450, 1977.

LASLO, Z. Project portfolio management: An integrated method for resource planning and scheduling to minimize planning/scheduling-dependent expenses. **International journal of project management**. p 609-618, 2010.

LEME, L. H. R. **Uma estratégia para apoiar o gerenciamento de riscos em um ambiente distribuído de desenvolvimento de software**. 2007. 108 f. Dissertação (Mestrado) - Universidade Estadual de Maringá, Paraná, 2007.

LYONS, T., SKITMORE, M. Project risk management in the queensland engineering construction industry: a survey. **International Journal of Project Management**, p. 51-61, 2004.

MACHLUP, F. **The production and distribution of knowledge in the United States**. Princeton: Princeton University Press, 1962.

MARTINS, G. A. **Manual para elaboração de monografias e dissertações**. São Paulo: Atlas, 2007.

MARTINS, G. A.; THEÓFILO, C. R. **Metodologia da investigação científica para ciências sociais aplicadas**. São Paulo: Editora Atlas, 2007.

MARTINS, L.; **Gestão profissional de projetos**, 2003. Disponível em http://www.ietec.com.br/ietec/techoje/techoje/gestaodeprojetos/2003/10/10/2003_10_10_0003.2xt/-template_interna. Acessado em 01/04/2011.

MENDES, J. J. M. **Sistema de apoio à decisão para planejamento de sistemas de produção tipo projecto**. 2003. 256 f. Dissertação (Mestrado) – Faculdade de engenharia da universidade do Porto, Porto, 2003.

MENESES, J. B. **Inspector: Um Processo de Avaliação de Progresso para Projetos de Software**. 2001. 201 f. Dissertação (Mestrado), Universidade Federal de Pernambuco, Recife, 2001.

MÜLLER, G. I.; RODRIGUES, A. G.; GÓMEZ, A.T. Um modelo baseado na busca tabu aplicado ao problema do escalonamento do job-shop com setup e data de entrega. **Hífen**, vol. 30, 2006.

MUNIZ, J. I. **Uso da tecnologia de grupos na automação industrial: Estudando problemas de Job-Shop**. 2009. 79 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica - Mecatrônica) - Universidade de Pernambuco, Pernambuco, 2009.

NAKAGAWA, M. **Introdução à controladoria: conceitos, sistemas, implementação**. São Paulo: Atlas, 1993.

NETO, J.; BOCOLI, F. Sucessosw = cmm2 + pmbok. *PMI Journal*, Publicação da Seção do PMI-RS. Número 5, Maio 2003. p. 2-11. Disponível em http://www.pmir.org/PMI20_Frame.htm. Acessado em 20/07/2011.

NORMAN, D. A. **Mental Models**. Hillsdale: Lawrence Erlbaum Associates, Inc., 1983.

NOWICKI, E.; SMUTNICKI, C. A. Fast taboo search algorithm for the job-shop problem. **Management Science**, vol. 42, p. 797-813, 1996.

PACHECO, R.F.; SANTORO, M.C. Proposta de classificação hierarquizada dos modelos de solução para o problema de job shop scheduling. **Gestão e produção**, Revista do Departamento de Engenharia de Produção, Universidade de São Carlos, p. 1-15, 1999.

PERMINOVA, O.; GUSTAFSSON, M.; WIKSTRÖM, K. Defining uncertainty in projects – a new perspective. **International Journal of Project Management**, p. 73-79, 2008.

PICH, M. T.; LOCH, C. H.; DE MEYER, A. On uncertainty, ambiguity, and complexity in project management. **Management Science**, Providence, v. 48, n. 8, p. 1008-1023, 2002.

PINTO, J. K.; KHARBANDA, O. P. How to fail at project management (without really trying). **Software Management**, v. 39, cap. 4, p. 45-53, 1996.

PMI (PROJECT MANAGEMENT INSTITUTE). **A guide to the project management body of knowledge: PMBOK guide**. 4. ed. Newton Square: Project Management Institute, 2008.

PRESSMAN, R. S. **Engenharia de software**. São Paulo: Makron Books, 2006.

RABECHINI JR., R.; CARVALHO, M. M. LAURINDO, F. J. B. Concepção de um programa de gerência de projetos em instituição de pesquisa. **Revista Valenciana d'Estudis Autonomics**, v.1, nº 20, p.1, 1999.

RABECHINI JR, R. **Competências e maturidade em gestão de projetos: uma perspectiva estruturada**. 2003. 274 f. Tese (Doutorado) – Escola Politécnica de Engenharia de Produção, Universidade de São Paulo, São Paulo, 2003.

REEVES, C. **Genetic algorithms and neighbourhood search**. Leeds: Evolutionary Computing, 1994.

SANDEEP, M. The accidental profession comes of age, 2002. Disponível em: <http://www.standards.org.au/STANDARDS/NEWSROOM/TAS/200206/PROJECT/PROJECT.HTM>. Acessado em 01/04/2011.

SCHMITZ, E. A.; ALENCAR, A. J.; VILLAR, C. B. **Modelos qualitativos de análise de risco para projetos de tecnologia da informação**. Rio de Janeiro: Brasport, 2006.

SEVERINO, A. J. **Metodologia do trabalho científico**. São Paulo: Cortez, 2007.

SHENHAR, A.; DVIR, D. **Reinventando gerenciamento de projetos: a abordagem diamante ao crescimento e inovação bem-sucedidos**. São Paulo: Makron Books, 2010.

SHIMAKURA, S. E. **Bioestatística** A. Departamento de Estatística, UFPR. Disponível em: <http://leg.ufpr.br/~shimakur/CE055/>. Acesso em: 14/04/2012.

SIEGEL, S.; CASTELLAN JR., N. J. **Estatística não-paramétrica para ciências do comportamento**. Porto Alegre: Artmed, 2006.

SILVA, M. M.; MARTINS, M. R.; RABECHINI JR, R. **Avaliação simultânea dos riscos de prazo e de custos em projetos**. São Paulo: PMI 11°. Seminário Internacional de Gestão de Projetos, 2011. Disponível em <
<http://pmisp.pmisp.org.br/11seminario/avalia%C3%A7%C3%A3o-simult%C3%A2nea-dos-riscos-de-prazo-e-de-custos-em-projetos-0>>. Acesso em 29.08.2011.

SOLÍS, J. F., CHÁVEZ, M.A.C. A reduced codification for the logical representation of job shop scheduling problems. **ICCSA 2004. LNCS**, vol. 3046, p. 553–562, 2004.

STECKE, K. E. A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems. **European Journal of Operational Research**, p. 24:369–375, 1986.

TAO, Y. A study of software development project risk management. **Management School**, p. 309-312, 2008.

TAVARES, M. G. P. **Cultura Organizacional**. Editora Quality Mark: Rio de Janeiro, 2002.

TURBAN, E. Decision-support and expert systems: management support systems. Macmillan Publ. New York: 1993.

VANLEHN, K. Problem-solving and cognitive skill acquisition. **Foundations of Cognitive Science – Cambridge: The MIT Press**, p. 526-579, 1989.

VASCONCELOS, R. V. J. C., FERREIRA FILHO, V. J. M. “Algoritmo Genético para o Problema de Scheduling de Projetos com Restrição de Recurso: Uma aplicação em Operações em Poços de Petróleo”, **Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional**, 2007.

VERGARA, S. C. **Projetos e relatórios de pesquisa em administração**. São Paulo: Atlas, 1998.

VIEIRA, E. Gerenciando projetos na era de grandes mudanças – Uma breve abordagem do panorama atual. **PMI Journal – PMI-RS**, p. 7-16, 2002.

VIRINE, L.; TRUMPER, M. **Project Decisions: The Art and Science**. Management Concepts, Inc, 2008.

WIDEMAN, R.M. **Project and program risk management – a guide to managing risks and opportunities**, Pennsylvania: Project Management Institute, 1992.

WOILER, S.; MATHIAS, W. F. **Projetos: planejamento, elaboração e análise**. 1. ed. São Paulo: Atlas S/A, 1996.

YIN, R. K. **Estudo de caso: planejamento e métodos**. 4. Ed. Porto Alegre: Bookman, 2010.

XIAOSONG, L.; SHUSHI, L.; WENJUN, C.; SONGJIANG, F. The application of risk matrix to software project risk management. **International Forum on Information Technology and Applications**. vol. 2, p. 480-483, 2009.