

ADRIANO FERNANDES
BÁRBARA P. EWERLING

SISTEMA DE PLANEJAMENTO FINANCEIRO E GESTÃO DE
MATERIAIS E EQUIPAMENTOS
(SISPLAN)

CURITIBA

2009

ADRIANO FERNANDES

BÁRBARA P. EWERLING

SISTEMA DE PLANEJAMENTO FINANCEIRO E GESTÃO DE
MATERIAIS E EQUIPAMENTOS
(SISPLAN)

Trabalho apresentado como requisito para obtenção de crédito na disciplina Trabalho De Conclusão De Curso (TI024) do curso de Tecnologia em Informática da Universidade Federal do Paraná.

Orientador: Professor MSc. Jaime Wojciechowski.

Curitiba

2009

TERMO DE APROVAÇÃO

ADRIANO FERNANDES

BÁRBARA P. EWERLING

SISTEMA DE PLANEJAMENTO FINANCEIRO E GESTÃO DE MATERIAIS E EQUIPAMENTOS (SISPLAN)

Trabalho de Conclusão de Curso aprovado como requisito para a obtenção do grau de Tecnólogo no Curso de Tecnologia em Informática da Universidade Federal do Paraná, Setor Escola Técnica, pela banca examinadora citada abaixo:

Orientador:

Professor MSc. Jaime Wojciechowski

Universidade Federal do Paraná, Setor Escola Técnica

Professor PhD Roberto Tadeu Raittz

Universidade Federal do Paraná, Setor Escola Técnica

Professora MSc. Sandramara S. K. de Paula Soares

Universidade Federal do Paraná, Setor Escola Técnica

Curitiba, ____ de _____ de 2009.

TERMO DE ACEITAÇÃO

A HFTI Tecnologia da Informação Ltda. declara para os devidos fins e para a Universidade Federal do Paraná que o produto final do Trabalho de Conclusão de Curso de Adriano Fernandes e Bárbara P. Ewerling, o software Sistema de Planejamento Financeiro e Gestão de Materiais e Equipamentos (SISPLAN) nos foi entregue, implantado e está sendo usado conforme acordado na ocasião da abertura do projeto (*Project Charter*).

Sem mais,

Eliane Dlugosz Wastowski
Gerente de Projetos de TI
HFTI Tecnologia da Informação

Ao Adriano, meu esposo.

À Bárbara, minha esposa.

Aos nossos pais e nossos irmãos.

À Dona Metilde.

Aos nossos amigos, afetos e ao Futuro.

AGRADECIMENTOS

Agradecemos por ainda estarmos inteiros, sadios, conscientes e praticamente ilesos.

Aos nossos companheiros, meu esposo e minha esposa, pela compreensão, por nos aturarmos em períodos difíceis e turbulentos e principalmente, por não nos deixar desistir.

À nossa família, pela paciência, por acreditarem em nós e pela boa vontade em nos ajudar em caminhos desconhecidos.

Ao orientador e professores, pelo acompanhamento em nossa jornada e principalmente, luz em momentos de escuridão.

À Universidade Federal do Paraná, pela estrutura para realizarmos o Projeto.

Aos nossos amigos, pelo companheirismo, pela calma e sabedoria.

Ao Cliente, por acreditar na competência da equipe para a realização do projeto.

*"Quem estuda e não pratica o que aprendeu
é como o homem que lava e não semeia."*

Provérbio Árabe

RESUMO

Nosso objetivo, com este documento, é descrever, detalhadamente, todos os aspectos, desde a idéia inicial, até sua aprovação e entrega ao cliente, de nosso Trabalho de Conclusão de curso.

Temos o propósito de implementar soluções eficientes que supram as deficiências nos controles de materiais e planejamento financeiro, conforme nos foi proposto.

O Sistema de Planejamento Financeiro e Gestão de Materiais e Equipamentos (SISPLAN) é a implementação de um aplicativo que permite planejar financeiramente as despesas com materiais, equipamentos e mão-de-obra de projetos específicos para o departamento de Engenharia, do nosso cliente, a Refinaria Presidente Getúlio Vargas (REPAR).

Seu desenvolvimento baseou-se na modelagem *Unified Modeling Language 2* (UML 2) para a especificação, documentação, e estruturação dos documentos, da notação de processos e, principalmente, a orientação a objetos.

O sistema foi projetado para uso *Desktop* e desenvolvido utilizando VB.NET (linguagem de programação), SQL Server (armazenamento de dados), MVC (arquitetura de desenvolvimento), UML 2 (padronização de modelagem) e Modelo Iterativo e Incremental (metodologia de desenvolvimento).

Palavras-chave: Gestão de Materiais. Planejamento Financeiro. Gestão de Equipamentos e Mão de Obra.

ABSTRACT

Our goal with this document is to describe in detail all aspects, from the initial idea until its approval and delivery to the customer, our Conclusion of course.

We propose to implement effective solutions that fix the deficiencies in the control of materials and financial planning, as has been proposed.

The System of Financial Planning and Management of Materials and Equipment (SISPLAN) are the implement of an application that enables you to plan financially the cost of materials, equipment and labor-intensive from projects specific to the engineering department, our client, Presidente Getúlio Vargas Refinery (REPAR).

Its development was based on modeling Unified Modeling Language 2 (UML 2) for the specification, documentation, and structuring of documents, the rating process and, especially, the object orientation.

The system is designed for Desktop use and developed using VB.NET (programming language), SQL Server (data storage), MVC (architecture development), UML 2 (standardization of modeling) and Model Iterative and Incremental (methodology development).

Keywords: Materials Management. Financial Planning. Equipment Management and Labor.

LISTA DE QUADROS

QUADRO 1 – RELAÇÃO DE NECESSIDADE X FUNCIONALIDADES	16
---	----

LISTAS DE SIGLAS E ABREVIATURAS

BW - *Business Warehouse*

ERP - *Enterprise Resource Planning*

GUI - Interface Gráfica do Utilizador (*Graphical User Interface*)

IBM - *International Business Machines*

MSc - Mestre (Grau Acadêmico - do latim *Magister Scientiæ*)

MVC - Modelo – Visualização – Controle (*Model – View – Controller*)

PETROBRAS - Petróleo Brasileiro S/A

PGP - Plano de Gerenciamento de Projetos

PhD - Doutor (Grau Acadêmico - do latim *Philosophiæ Doctor*)

REPAR - Refinaria Presidente Getúlio Vargas

SAP - *System Analysis and Program Development*

SISPLAN - Sistema de Planejamento Financeiro e Gestão de Materiais e Equipamentos

SQL - Linguagem de Consulta Estruturada (*Structured Query Language*)

UFPR – Universidade Federal do Paraná

UML - *Unified Modeling Language*

VB - *Visual Basic*

XP - Experiência (*eXPerience*)

LISTAS DE SÍMBOLOS

® - Marca Registrada

™ - Trade Mark (Marca Registrada)

SUMÁRIO

1	INTRODUÇÃO	13
2	ESPECIFICAÇÃO DO PROJETO	14
2.1	DESCRIÇÃO DO CLIENTE	14
2.2	DESCRIÇÃO DO PROBLEMA	14
2.2.1	Necessidades e funcionalidades	15
2.3	DESCRIÇÃO DA SOLUÇÃO	16
3	METODOLOGIA PARA O DESENVOLVIMENTO DO PROJETO	17
3.1	FASES DA METODOLOGIA	18
3.1.1	Levantamento de Requisitos	19
3.1.2	Análise	19
3.1.3	Implementação	21
3.1.4	Testes	21
3.1.5	Implantação	22
3.2	GERENCIAMENTO DO PROCESSO DE SOFTWARE	22
3.3	TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO FUNCIONAL	24
3.4	TÉCNICAS E FERRAMENTAS PARA MODELAGEM DE DADOS	27
3.5	FERRAMENTAS PARA IMPLEMENTAÇÃO	27
3.5.1	Requisitos determinados pelo cliente	27
3.5.2	Métodos de trabalho adotado pela equipe	28
3.6	TÉCNICAS E FERRAMENTAS PARA PROCESSO DE HOMOLOGAÇÃO	29
3.6.1	Homologação interna	29
3.6.2	Homologação pelo cliente	29
4	PROCESSO DE DESENVOLVIMENTO DA APLICAÇÃO SISPLAN	30
4.1	APLICAÇÃO DA METODOLOGIA ADOTADA	30
4.2	APLICAÇÃO DO GERENCIAMENTO DO PROCESSO DE SOFTWARE	31
4.3	TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO FUNCIONAL	31
4.4	APLICAÇÃO DAS TÉCNICAS E FERRAMENTAS DE MODELAGEM DE DADOS	32
4.5	APLICAÇÃO DAS FERRAMENTAS DE IMPLEMENTAÇÃO	33
4.6	APLICAÇÃO DAS TÉCNICAS DE HOMOLOGAÇÃO INTERNA	33
5	HOMOLOGAÇÃO PELO CLIENTE	35
6	ESTRATÉGIAS DE IMPLANTAÇÃO	36
7	CONSIDERAÇÕES FINAIS	37
	REFERÊNCIAS BIBLIOGRÁFICAS	39
	GLOSSÁRIO	41
	APÊNDICE 1 – PLANO DE GERENCIAMENTO DO PROJETO (PGP)	42
	APÊNDICE 2 – DIAGRAMA DE CASOS DE USO	43
	APÊNDICE 3 – ESPECIFICAÇÃO DOS CASOS DE USO	44
	APÊNDICE 4 – DIAGRAMA DE CLASSES	45
	APÊNDICE 5 – DIAGRAMAS DE SEQUÊNCIA	46
	APÊNDICE 6 – DIAGRAMA ENTIDADE RELACIONAMENTO	47
	APÊNDICE 7 – PADRÕES DE NOMENCLATURA DE DESENVOLVIMENTO	48
	APÊNDICE 8 – CÓDIGOS FONTE	49

1 INTRODUÇÃO

A disciplina Trabalho De Conclusão De Curso (TI024) tem como objetivo pôr em prática o conhecimento que adquirimos, ao longo do curso de Tecnologia em Informática, sobre concepção e gerenciamento de software. O método escolhido para a avaliação é o desenvolvimento de um projeto, de conteúdo proposto pela equipe, no nosso caso em específico, com colaboração do orientador e professores que auxiliarão na evolução do processo.

A escolha do tema foi, além de criteriosa, benéfica para o cliente e para a equipe. Resolvemos um problema / dor de cabeça do cliente, com a nossa necessidade de cumprir o objetivo da disciplina. Nossa escolha chama-se Sistema de Planejamento Financeiro e Gestão de Materiais e Equipamentos (SISPLAN) e trata-se de um programa que torna possível o acompanhamento do processo de aquisição de materiais e equipamentos e gerar relatórios com os status de compras, estoques e necessidades.

A evolução financeira do projeto e o processo de compra são acompanhados com base em informações extraídas do sistema ERP SAP/R3 e/ou do sistema de Datawarehouse BW. A modelagem do sistema foi construída sobre a UML 2.

Em resumo, é um aplicativo que permite planejar financeiramente despesas com materiais, equipamentos e mão-de-obra de projetos.

Primeiramente, o projeto será apresentado e após, será introduzida a especificação de sua documentação, com os métodos utilizados para a elaboração do projeto e por fim, será demonstrado o material que usamos com base para nosso trabalho.

2 ESPECIFICAÇÃO DO PROJETO

2.1 DESCRIÇÃO DO CLIENTE

A HFTI Tecnologia da Informação Ltda. atua desde 1991 oferecendo soluções e serviços na área de Tecnologia da Informação. Com sede em Curitiba e filial no Rio de Janeiro, a empresa opera presencialmente junto a clientes em várias regiões do país.

A empresa presta serviço de consultoria SAP na Refinaria Presidente Getúlio Vargas (REPAR) unidade pertencente ao grupo PETROBRAS S.A., situada em Araucária/PR. Uma das atividades da empresa é desenvolver ferramentas de apoio que supram as deficiências e/ou limitações encontradas na extração de informações e planejamento de projetos de engenharia no sistema ERP SAP/R3.

2.2 DESCRIÇÃO DO PROBLEMA

Uma vez, estando com o cliente, nos foram relatadas as dificuldades em fazer o planejamento financeiro e a gestão de materiais. As operações envolvidas eram dificultosas e até, de certo modo, muito trabalhosas e braçais e sem automação alguma. Uma verdadeira mineração de dados.

Para realizar o planejamento financeiro eram utilizadas várias planilhas provenientes dos diversos setores (especialidades) envolvidos e, cada um informava a distribuição anual para as contas

(documentos) que iria utilizar. Após isso, tudo era colocado em um único arquivo contendo diversas planilhas e, então, estas eram sumarizadas em outras duas ou três de modo a obterem mais de uma visão do planejamento total. Na seqüência e posterior ao processo de análise, estes custos eram distribuídos mensalmente, utilizando outro arquivo para isso. Até mesmo os links criados para automatizar o arcaico processo de cálculo eram frágeis e freqüentemente tinham que ser revisados.

Na gestão de materiais e equipamentos o controle, acompanhamento e organização das listas de compra também eram realizados em planilhas eletrônicas oriundas de diversas outras. Além disso, para garantir um histórico sobre as ocorrências durante o processo de aquisição, eram criadas pastas com emails, documentos e anotações sem nenhum critério de organização.

A partir do momento que definimos exatamente o problema a ser resolvido, partimos para uma solução prática, ou seja, relacionamos os requisitos de alto-nível, as necessidades dos usuários, as funcionalidades do sistema e o principal, projetamos e implementamos o desenvolvimento de um software.

Começaremos com a visão geral do sistema, após, documentaremos o amplo ambiente de processos envolvidos no sistema, e então, forneceremos para todos os envolvidos, o produto de nosso trabalho e documentação compreensível deste e suas funcionalidades.

2.2.1 Necessidades e funcionalidades

Veja no quadro abaixo, a relação necessidade x funcionalidade que o cliente relatou como de extrema importância a ser implementado na solução.

NECESSIDADE		BENEFÍCIO
Carregar as informações extraídas do SAP e do BW		Crítico
ID. FUNC.	DESCRIÇÃO DAS FUNCIONALIDADES	ATORES
F 1.1	Importar dados do BW	Administrador
F 1.2	Importar dados do SAP/R3	Administrador

QUADRO 1 – RELAÇÃO DE NECESSIDADE X FUNCIONALIDADES

FONTE: Os autores (2009)

2.3 DESCRIÇÃO DA SOLUÇÃO

Considerando situação que o cliente nos apresentou, foi determinado que o sistema a ser desenvolvido deveria permitir o planejamento financeiro das despesas com materiais, equipamentos e mão-de-obra de um projeto específico para o departamento de engenharia da REPAR.

O sistema torna possível, também, acompanhar o processo de aquisição destes materiais e equipamentos e gerar relatórios com os status das compras, estoque e necessidades.

A evolução financeira do projeto e o processo de compra são acompanhados com base em informações extraídas do sistema ERP SAP/R3 e/ou do sistema de Datawarehouse BW (ambos de propriedade da empresa alemã SAP). Estas informações restringem-se apenas aos usuários definidos pela coordenação do projeto.

Com estes dados em mãos, o cliente fez uma avaliação positiva e assim, iniciou-se um processo de maior refinamento das funcionalidades e implementação, conforme descrição no Plano de Gerenciamento de Projetos (PGP).

3 METODOLOGIA PARA O DESENVOLVIMENTO DO PROJETO

No meio do ambiente de desenvolvimento de projetos, considera-se que a metodologia adotada para a execução do envolve todo o ciclo de vida do plano em questão. Isso inclui as regras criadas, os padrões adotados, ferramentas que serão utilizadas, e todos os pontos relevantes para a definição do processo de desenvolvimento completo.

Para o processo de criação do SISPLAN, convencionamos o uso do Modelo Iterativo e Incremental como metodologia de desenvolvimento de software, por ter etapas bem definidas e cada uma é composta de objetivos alinhados. Conforme a Wikipédia, a definição de desenvolvimento incremental e iterativo é a que segue:

Desenvolvimento Incremental é uma estratégia de planejamento estagiado em que várias partes do sistema são desenvolvidas em paralelo e integradas quando completas. [...] A alternativa ao desenvolvimento incremental é desenvolver todo o sistema com uma integração única.

Desenvolvimento iterativo é uma estratégia de planejamento de retrabalho em que o tempo de revisão e melhorias de partes do sistema é pré-definido. [...] A saída de uma iteração é examinada para modificação, e especialmente para revisão dos objetivos das iterações sucessivas.¹

Ao que consta na literatura, um processo de desenvolvimento de software, deve, necessariamente, ter as fases que seguem na citação e a partir desta, observa-se a razão de adotarmos o

¹ **Desenvolvimento Iterativo e Incremental.** Disponível em: <http://pt.wikipedia.org/wiki/Desenvolvimento_iterativo_e_incremental> Acesso em 24 set. 2009.

desenvolvimento incremental e iterativo como padrão durante o processo de software.

...As principais fases de um processo de software são:

1. Especificação de Requisitos: tradução da necessidade ou requisito operacional para uma descrição da funcionalidade a ser executada.
2. Projeto de Sistema: tradução destes requisitos em uma descrição de todos os componentes necessários para codificar o sistema.
3. Programação (Codificação): produção do código que controla o sistema e realiza a computação e lógica envolvida.
4. Verificação e Integração (Verificação): verificação da satisfação dos requisitos iniciais pelo produto produzido.²

3.1 FASES DA METODOLOGIA

Tomando por base o desenvolvimento incremental e iterativo e selecionando os seus aspectos, convencionamos que as seguintes fases deveriam ser seguidas e executadas como metodologia de desenvolvimento, na ordem apresentada:

1. Levantamento de requisitos;
2. Análise;
3. Implementação;
4. Teste;
5. Implantação.

² **O Processo de Software.** Disponível em: <http://www.macoratti.net/proc_sw1.htm> Acesso em 24 set. 2009.

3.1.1 Levantamento de Requisitos

Tomando como base a solicitação inicial de nosso cliente, todos os requisitos do sistema foram listados e seus limites definidos nesta fase. Fizemos, através desta etapa, a identificação das funcionalidades necessárias e estas, foram listadas e descritas e desta forma, proporcionando a base para o desenvolvimento do detalhamento e o avanço para as fases seguintes.

Neste momento, é feita a elaboração e descrição dos casos de uso, as regras de negócio são bem definidas e um protótipo do sistema é elaborado.

Fizemos um protótipo utilizando o Microsoft® Office Access® que “é um sistema de gerenciamento de banco de dados da Microsoft [...] que combina o Microsoft Jet Database Engine com uma interface gráfica do utilizador (*graphical user interface* [GUI]). Ele permite o desenvolvimento rápido de aplicações que envolvem tanto a modelagem e estrutura de dados como também a interface a ser utilizada pelos usuários.”³

Apresentamos nosso trabalho para avaliação do cliente com o objetivo de que ambas as partes compreendessem as partes do objetivo e do escopo do sistema. Com a aprovação dos casos de uso e do protótipo, a próxima fase pode ser iniciada.

3.1.2 Análise

Finalizamos a construção dos casos de uso e tão logo, foi iniciada a análise do sistema e este é o momento que definimos sua

³ **Microsoft Access.** Disponível em: <http://pt.wikipedia.org/wiki/Microsoft_Access> Acesso em 15 set. 2009.

arquitetura. Na definição da arquitetura, preferimos uma que fizesse a divisão do desenvolvimento do sistema em camadas, para a melhor separação e centralização das responsabilidades de cada camada.

Nossa escolha foi pelo modelo de três camadas o Modelo – Visualização – Controle (*Model – View – Controller* [MVC]). Este consiste em três camadas físicas e o aplicativo é dividido de modo que a Camada de Negócios (Modelo) fique no centro das três camadas físicas, entre a Camada de Apresentação (Visualização) e da Camada de Controle (Controle).

A arquitetura MVC não é nova e foi desenvolvida com objetivo de descrever as seqüências de entrada, processamento e saída para o modelo de interação com o usuário. Usando este molde, facilmente mapeamos estes conceitos no domínio de aplicações multicamadas e como ele gerencia múltiplos visualizadores usando o mesmo modelo é fácil manter, testar e atualizar sistemas múltiplos.

A Camada de Negócios (Modelo) – identificada no projeto com a sigla BLL de Business Logic Layer – é onde está o código que obtém os dados retornados pela DAL e os repassa ao cliente de uma forma intuitiva, efetuando a validação lógica conforme as regras do negócio e verificando se os dados estão no formato adequado.

A Camada de Apresentação (Visualização) – identificada no projeto com a sigla UI de User Interface – é onde está o código que define o que será efetivamente apresentado ao usuário, efetuando a formatação dos dados, navegação e ajuda ao usuário final.

A Camada de Controle (Controle) – identificada no projeto com a sigla DAL de Data Access Layer – é onde está o código que trata da obtenção, tratamento e retorno dos dados salvos na camada de armazenamento de dados.

Nesta fase também foram desenvolvidos os diagramas essenciais do sistema, como o diagrama de classes, seqüência e relacional. Também foi feita a modelagem do banco de dados.

3.1.3 Implementação

O objetivo desta fase é a implementação do sistema modelado, conforme descrito documentação criada e definida nas fases anteriores. Neste ponto, aplicamos nosso conhecimento sobre as regras do negócio e dividimos o sistema em módulos, para que este fosse perfeitamente integrado no sistema com um todo mais tarde.

Fizemos análise detalhada dos casos de uso e da série de diagramas e desta maneira, implementamos o sistema conforme este roteiro. Em possibilidade de inconsistências na documentação em relação ao executado, fazíamos os ajustes necessários e quando necessário, alteramos a documentação, em novas versões, para adaptarmos à necessidade identificada.

3.1.4 Testes

Convencionamos, conforme a metodologia adotada, que os testes de cada módulo deveriam ser feitos pelos integrantes que não haviam programado o determinado módulo. Uma medida simples e eficiente, cada um testa o módulo que o outro membro desenvolveu e documenta os resultados.

Antes de colocar o sistema para uso do cliente, uma equipe ou um grupo de usuários deve realizar os testes do cliente, visando o

correto funcionamento do sistema e também verificar se o produto final foi o que o cliente esperava.

3.1.5 Implantação

Após finalizar o processo de desenvolvimento, obtivemos como produto final, uma versão funcional do software e finalmente, o sistema deverá ser implantado no cliente, levando em conta o ambiente necessário e os pré-requisitos definidos inicialmente para seu correto funcionamento.

3.2 GERENCIAMENTO DO PROCESSO DE SOFTWARE

“Um processo de software é formado por um conjunto de passos de processo parcialmente ordenados, relacionados com artefatos, pessoas, recursos, estruturas organizacionais e restrições, tendo como objetivo produzir e manter os produtos de software finais requeridos.”⁴

“De modo geral, considera-se que os objetivos da Engenharia de Software são o aprimoramento da qualidade dos produtos de software e o aumento da produtividade [...] além do atendimento aos requisitos de eficácia e eficiência, ou seja, efetividade.”⁵

⁴ **Automação no Gerenciamento do Processo de Engenharia de Software.** Disponível em <<http://www.labes.ufpa.br/quites/teaching/2006/TopicosES/ProcessoEngenhariaSoftware.pdf>> Acesso em 10 set. 2009.

⁵ REZENDE, D. A. **Engenharia de software e sistemas de informação.** 3 ed. Rio de Janeiro: Brasport, 2005. p. 02.

Engenharia de software é metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: processo (roteiro) dinâmico, integrado e inteligente de soluções tecnológicas; adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes; efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos; fundamentação na Tecnologia da Informação disponível, viável, oportuna e personalizada; planejamento e gestão de atividades, recursos, custos e datas.⁶

Em resumo, o conjunto de atividades, práticas e métodos usados para desenvolver e manter um software e suas ramificações é o que define um processo de software. Todos estes quesitos rumam para a mesma direção: proceder com um processo com qualidade para resultar em um produto com qualidade. Há de ser ressaltado que, dentre outras necessidades, uma das mais importantes, é a de gerência.

A análise e o projeto, os algoritmos, a estruturação de dados e linguagens devem ser tratadas com técnicas e ferramentas específicas, mas a gerência do processo deve se utilizar de conceitos e métodos análogos aos de outras áreas da empresa. As ferramentas para controle gerencial e avaliação também são as mesmas.

Conforme Pressman, a qualidade de um software é definida como "...Conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente definido..."⁷

A gerência do processo estabelece a infra-estrutura para suportar e guiar os trabalhos dos diversos projetos de maneira uniforme. A gerência inclui:

⁶ REZENDE, D. A. **Engenharia de software e sistemas de informação**. 3. ed. Rio de Janeiro: Brasport, 2005. p. 05.

⁷ PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw-Hill Interamericana do Brasil, 2006. p. 52.

1. Definir o processo: estabelecer padrão para implementação, avaliação e melhoria das tarefas;
2. Executar o processo: definir métodos e técnicas usadas para produzir produtos com qualidade;
3. Coletar e analisar dados: tratar as medições realizadas em produtos e processos de software, e o uso desses dados;
4. Controlar o processo: estabelecer mecanismos para certificar o desempenho do processo; monitorar e fazer ajustes onde melhorias forem necessárias.

O gerenciamento do processo de desenvolvimento no SISPLAN foi definido no PGP, com detalhamento dos objetivos, descrição de problema e as ferramentas utilizadas para atingi-los. E para um controle geral, desenvolvemos um cronograma composto com datas limites e tarefas a serem executadas, atualizado periodicamente, para um controle eficaz e cumprimento, no prazo estipulado, das atividades da equipe.

3.3 TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO FUNCIONAL

Conforme José Carlos Macoratti, “Os requisitos expressam as características e restrições do produto de software do ponto de vista de satisfação das necessidades do usuário, [...] sendo a parte mais crítica e propensa a erros no desenvolvimento de software.”⁸

⁸ **A Gestão de Requisitos.** Disponível em <http://imasters.uol.com.br/artigo/3860/des_de_software/a_gestao_de_requisitos/> Acesso em 11 set. 2009.

E antes de partir para a especificação dos requisitos, há de se explicar o que é um requisito funcional e um requisito não-funcional.

...Os requisitos funcionais são a descrição das diversas funções que clientes e usuários querem ou precisam que o software faça. Eles definem a funcionalidade desejada do software.

Requisitos não-funcionais são as qualidades globais de um software, como manutenibilidade, usabilidade, desempenho, custos e várias outras. Normalmente estes requisitos são descritos de maneira informal, de maneira controversa [...] e são difíceis de validar...⁹

Segundo Wazlawick, “a modelagem funcional especifica as funções externas do sistema, e descreve as Operações do Sistema como entradas, e as Consultas do Sistema como saídas.”¹⁰

Como metodologia adotada para a realização do processo de desenvolvimento de software orientado a objetos, foi utilizado o modelo iterativo e incremental por ser uma ferramenta bem estruturada e que condiz com nossos objetivos, ou seja, supre a necessidade de diagramas e ela é totalmente compatível com a metodologia adotada para a modelagem funcional.

Para o SISPLAN, a modelagem funcional tem início com o diagrama de casos de uso e de seu detalhamento. Cada diagrama é compreendido de pré-condições, funcionamento, fluxos alternativos e exceções. Estes foram construídos em cima das Regras de Negócio, conforme orientações do cliente.

Procuramos fazer a modelagem funcional através da implementação da *Unified Modeling Language 2 (UML 2)*, amplamente conhecida no meio de desenvolvimento de sistemas, pela padronização dos diagramas, de fácil entendimento a todos.

⁹ **A Gestão de Requisitos.** Disponível em <http://imasters.uol.com.br/artigo/3860/des_de_software/a_gestao_de_requisitos/> Acesso em 11 set. 2009.

¹⁰ WAZLAWICK, R. S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos.** 1 ed. São Paulo: Campus, 2004. p. 102.

A UML é uma notação gráfica usada para criar modelos orientados a objetos para análise e projeto de software orientado a objetos. Não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida ou como projetar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos.

O modelo precursor da notação surgiu em 1994, publicado com o título Unified Method 0.8. A partir de então,

...Uma revisão ampla levou à UML 2.0. que apresenta grandes ampliações e mudanças em comparação à versão 1.x. É o caso, por exemplo, dos diagramas de atividade e de seqüência. Além disso, o metamodelo, ou seja, o modelo de UML de especificação de UML foi inteiramente revisado. A OMG publicou os primeiros documentos em agosto de 2003. Em outubro de 2004, uma versão corrigida desse documento foi preparada e publicada no primeiro trimestre de 2005...¹¹

Basicamente, a UML permite que desenvolvedores visualizem os produtos de seu trabalho em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica. É uma notação independente de processos.

Os objetivos da UML são: especificação, documentação, e estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação. A UML é um modo de padronizar as formas de modelagem.

Já, a modelagem não-funcional do sistema foi descrita nos casos de uso, como as regras de negócio, que descrevem restrições, regras, e requisitos mínimos e requisitos globais para o funcionamento e bom desempenho do sistema.

¹¹ BALZERT, H. **UML 2: compacto**. 1 ed. Rio de Janeiro: Elsevier, 2008. p. 2.

3.4 TÉCNICAS E FERRAMENTAS PARA MODELAGEM DE DADOS

A modelagem de dados é a atividade que especifica em detalhes as estruturas de dados e regras de negócios para a montagem de um banco de dados íntegro, o qual compõe o sistema desenvolvido.

Modelar implica em construir modelos. No caso do SISPLAN, foi utilizado o Microsoft® Office Visio® 2003, que é um programa de diagramação técnica, que auxilia na ilustração de processos e sistemas através de diagramas. O Visio pode ser utilizado para gerar diagramas de diversos tipos, como organogramas, fluxogramas, diagramas de redes, e dentre outros, modelagem de dados usando UML, que é o nosso caso.

3.5 FERRAMENTAS PARA IMPLEMENTAÇÃO

3.5.1 Requisitos determinados pelo cliente

Para a realização de nosso trabalho, o cliente propôs algumas condições a serem seguidas. Segundo estas condições, que constam no PGP, o ambiente para desenvolvimento deveria ser compatível com a plataforma Microsoft. Optamos pelo Microsoft® .NET Framework™ que é um componente do Windows que oferece suporte à criação e execução de aplicativos e visa uma plataforma única para desenvolvimento e execução destes. Todo e qualquer código gerado para .NET, pode ser executado em qualquer dispositivo que possua um framework de tal plataforma.

O .NET fornece ambiente de programação orientado a objetos consistente; ambiente que promove segurança para execução de código que minimiza conflitos de implantação e versionamento de software; e cria comunicação padrão, para garantir que códigos baseados no .NET Framework possam se integrar a qualquer outro código.

Outra condição proposta pelo cliente foi a utilização do SQL Server para gerenciamento do banco de dados por já ser um sistema conhecido e já acreditado pela empresa.

O Microsoft® SQL Server™ 2005 é um sistema gerenciador de Banco de Dados Relacional (SGBD) criado pela Microsoft. Ele é um Banco de dados robusto e usado por sistemas corporativos dos mais diversos portes.

Dentre suas principais ferramentas consta que é um banco de dados relacional; com serviços de replicação de dados, integração com warehouses, análise e relatórios; tem ferramentas de gerenciamento e ferramentas de desenvolvimento, com total integração com o MS Visual Studio 2005.

3.5.2 Métodos de trabalho adotado pela equipe

Ao nos comprometermos com o cliente e suas condições, tivemos que adotar certas técnicas e medidas para o dinamismo de nosso trabalho.

Numa destas medidas concluímos que junto com o Visual Studio teríamos que adotar uma linguagem para programação. E em nosso projeto, escolhemos utilizar o Visual® Basic™, que é uma linguagem de programação totalmente orientada a objetos e com

suporte total a UML, criada pela Microsoft e distribuída com o Microsoft® Visual Studio®.NET.

Já para a adoção do SQL Server, levamos em consideração a redução do tráfego de rede e a um bom desempenho da velocidade de execução.

3.6 TÉCNICAS E FERRAMENTAS PARA PROCESSO DE HOMOLOGAÇÃO

3.6.1 Homologação interna

O método utilizado para homologação interna do sistema foi o de testes com base nos casos de uso desenvolvidos. Os testes eram alternados: os módulos desenvolvidos por um integrante da equipe eram testados pelo outro integrante e este processo era mútuo. Os resultados eram documentados e guardados com marcação de êxito ou ajustes a serem feitos. Este procedimento foi adotado pela nossa equipe por ser muito eficiente na busca de falhas.

3.6.2 Homologação pelo cliente

Esta etapa do desenvolvimento ainda não pôde ser realizada devido a um processo de manutenção da refinaria. É um procedimento interno do cliente e por esta razão, ela encontra-se fechada para intervenções externas, não tivemos oportunidade de completar esta fase do projeto.

4 PROCESSO DE DESENVOLVIMENTO DA APLICAÇÃO SISPLAN

4.1 APLICAÇÃO DA METODOLOGIA ADOTADA

Procuramos manter a metodologia de desenvolvimento que foi definida no planejamento do SISPLAN em todas as fases do projeto, no entanto, tivemos que fazer algumas adaptações para nosso caso específico.

A fase de levantamento de requisitos foi realizada com êxito. Nesta, foi elaborado o protótipo do sistema e a entregamos para o cliente e este, por sua vez, aprovou esta fase. Complementamos esta ainda, com casos de uso ilustrados com as telas correspondentes.

Quanto à diagramação do sistema, foram feitos de modo a representar a implementação do modo que ela seria feita no sistema. Ao longo do período que o projeto foi desenvolvido, constatou-se, através de testes, checagem da consistência e integridade e até limitação da plataforma de desenvolvimento e gerenciador do banco de dados, que alguns itens poderiam ficar erroneamente representados nos casos de uso, diagramas e até na modelagem de dados. Com esta conclusão, alguns itens foram reformulados para a documentação condizer ao produto final.

Sobre os testes de sistema, eram realizados periodicamente, ao final da construção de cada módulo e os resultados eram analisados, em caso positivo, dávamos continuidade à construção e em caso negativo, ajustes eram feitos conforme a necessidade.

4.2 APLICAÇÃO DO GERENCIAMENTO DO PROCESSO DE SOFTWARE

Começamos a fase de especificação arquitetando o PGP com todos os seus itens, a especificação, o planejamento, condições, estimativas, dentre outros. No entanto, alguns requisitos não puderam ser cumpridos conforme o cronograma em função de alguns imprevistos no meio do caminho e fundamentalmente a relação tempo x trabalho a ser feito.

A implementação e os testes correram conforme o planejado e não há pontos com mais ou menos dificuldades, todos os processos foram considerados de mesma complexidade.

Quanto à implantação e homologação feita pelo cliente, não pode ser feita ainda por conta de algumas manobras de segurança e manutenção que o mesmo planejou para implantar na mesma época que tínhamos em nosso cronograma para entregar o sistema.

4.3 TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO FUNCIONAL

O início da modelagem de nosso sistema foi criada no Microsoft Visio e mais tarde, conforme as necessidades, esta foi modificada. Em seguida, iniciamos o detalhamento dos casos de uso junto com o levantamento de requisitos e a criação do protótipo, feito no MS Access com linguagem VBA. Estes documentos contêm os casos de uso ilustrados com as telas criadas no protótipo.

A modelagem funcional requer o diagrama de classes e podemos afirmar que este é o mais importante e o mais utilizado da UML e é usado como base para vários outros diagramas.

Para atender as necessidades da modelagem funcional foi desenvolvido também o diagrama de classes, que é o diagrama mais utilizado e o mais importante da UML, servindo de apoio para a maioria dos outros diagramas. A seguir, desenvolvemos os diagramas de seqüência, baseados nos casos de uso e nos diagramas de classe. O diagrama de seqüência cuida do fluxo de mensagens entre os objetos do processo do ponto de vista temporal. Comumente, fundamenta-se um caso de uso definido pelo diagrama de mesmo nome e o apóia no diagrama de classes para determinar os objetos das classes envolvidas em um processo.

4.4 APLICAÇÃO DAS TÉCNICAS E FERRAMENTAS DE MODELAGEM DE DADOS

Iniciamos a o planejamento do banco de dados com a modelagem do diagrama de entidades e relacionamentos (DER) e buscamos fazer a modelagem inteira antes do início do desenvolvimento. Entretanto, durante o desenvolvimento do sistema, alguns detalhes tiveram que ser modificados e isso fez com que o DER tivesse suas modificações também.

4.5 APLICAÇÃO DAS FERRAMENTAS DE IMPLEMENTAÇÃO

As ferramentas definidas no desenho do projeto para implementação do sistema foram utilizadas visando a reutilização de classes, conforme o conceito de orientação a objetos aconselha.

Antes de começarmos com o desenho do projeto, nos reunimos junto ao cliente e este nos definiu as histórias de usuário, que para nós, são as regras de negócio. Após esta fase, construímos todos os diagramas UML e definimos os métodos de nosso sistema conforme instruções da literatura e de nosso professor orientador.

Algumas definições foram feitas de acordo com as limitações da plataforma de desenvolvimento e/ou do gerenciador de banco de dados. Com estas restrições em mãos, tivemos que criar soluções sem qualquer definição já planejada. Depois, estas foram incluídas nas respectivas documentações.

Todos os documentos foram gerados a partir do planejamento do sistema foram armazenados remotamente em um servidor público de acesso individual. Foi de grande utilidade principalmente pelo histórico feito com este processo.

4.6 APLICAÇÃO DAS TÉCNICAS DE HOMOLOGAÇÃO INTERNA

Durante cada etapa de desenvolvimento, o programador realizou os testes iniciais no módulo desenvolvido, conhecido como teste unitário. Após o término deste processo, foi preparada uma lista de checagem para testar a funcionalidade do módulo. Esta lista considerou os parâmetros definidos pelos casos de uso; a relação das

configurações de tela com o protótipo; e os dados alterados no banco e em diretórios físicos.

Os testes foram realizados no sistema de troca, alternamos a realização das tarefas de teste e desenvolvimento entre os membros da equipe, e assim, garantimos que o desenvolvedor não realize os testes no código-fonte gerado por ele mesmo. Os ajustes e um último ciclo de testes foram realizados, tendo todos os itens sido atendidos conforme esperado.

5 HOMOLOGAÇÃO PELO CLIENTE

Para esta importante etapa do processo, a homologação do sistema pelo cliente ser realizada, previamente, iremos dispor o sistema em um servidor de testes, para possibilitar a verificação de alguma inconsistência do sistema no ambiente do cliente.

Solicitaremos ao cliente que faça testes de desempenho e funcionalidade e que nos reporte, com detalhes, tanto os resultados positivos, quanto os resultados negativos, para ajustes.

Caso tenhamos que alterar o sistema por termos obtido um resultado negativo, o processo de testes será refeito. Consideremos este processo como um ciclo. Nossa expectativa é que este ciclo seja percorrido até três vezes, porém, se resultados positivos forem obtidos antes, não terá mais repetição. Caso contrário, a repetição será feita até obtermos uma resposta positiva.

6 ESTRATÉGIAS DE IMPLANTAÇÃO

Considerando o ambiente cliente e o sistema desenvolvido, ou seja, o produto final, indicamos alguns requisitos a serem atendidos para a sua implantação. São os que seguem:

1. Para o servidor de banco de dados: este deverá ter o SQL Server 2005 instalado, juntamente com todas as suas atualizações fornecidas pelo fabricante.

2. Para as estações de trabalho: todas devem ter como sistema operacional o Microsoft ® Windows® XP ou superior, integralmente atualizado e o pacote Microsoft ® Office 2003 ou superior com as últimas atualizações instaladas.

3. Para a primeira utilização do sistema: dois procedimentos devem ser realizados antes que o sistema seja executado pela primeira vez. O administrador do sistema deve, necessariamente, cadastrar o login e a senha do usuário, juntamente com o mesmo.

Tanto as versões do Windows com as do Office são softwares proprietários da fabricante Microsoft.

7 CONSIDERAÇÕES FINAIS

Conforme citamos anteriormente, a homologação do cliente ainda não pode ser realizada. Por este motivo, não temos uma palavra final, uma avaliação do cliente sobre o produto final de nosso projeto. A previsão de entrega é para dia 30 de setembro de 2009.

Enquanto construíamos nosso projeto, vimos novas possibilidades, novos pontos de vista, novas funcionalidades a serem adicionadas e até mesmo, complementar algumas funcionalidades já existentes. Como nosso planejamento já havia sido bem estruturado com o cronograma que já não permitia mais acréscimos de tempo e as definições já haviam sido aprovadas pelo cliente, tivemos que deixar estas implementações para o futuro.

Com estas situações em mãos, faremos a proposta ao cliente de uma nova versão com as seguintes possibilidades de implementações futuras:

- Documento de Ajuda do sistema
- Módulo de Planejamento
 - Cenário – criar sistema que permita copiar todos os documentos gerados para um determinado cenário, de modo, que o novo cenário gerado herde todos os documentos, estruturas, valores anuais e mensais do cenário escolhido.
 - Importar dados (SAP / BW) dos valores realizados nos projetos e implementar comparação PLANEJADO X REALIZADO.
 - Para o Planejamento Analítico, desenvolver um sistema de exportação para Excel no formato em que os dados são visualizados e de acordo com o grupo selecionado.

- Grupo Engenharia – implementar recurso *drag and drop* na *treeview* para permitir mais visualizações dos valores planejados.

- Módulo de Materiais

- Importar os relatórios de compra e cadastro de material (SAP e BW).

- Elaborar relatórios e gráficos referentes às situações das compras.

- Relatório de histórico por requisição de compra.

REFERÊNCIAS BIBLIOGRÁFICAS

1. **A Gestão de Requisitos.** Disponível em: <http://imasters.uol.com.br/artigo/3860/des_de_software/a_gestao_de_requisitos/> Acesso em 11 set. 2009.
2. **Artigos SQL Server e VB.NET.** Disponível em: <http://www.arquivodecodigos.net/arquivo/tutoriais/vb_net.php> Acesso em 2009 (durante toda duração do projeto).
3. **Artigos SQL Server e VB.NET.** Disponível em: <<http://www.vbmania.com.br/pages/index.php>> Acesso em 2009 (durante toda duração do projeto).
4. **Artigos SQL Server e VB.NET.** Disponível em: <<http://social.msdn.microsoft.com/Forums/pt-BR/>> Acesso em 2009 (durante toda duração do projeto).
5. **Automação no Gerenciamento do Processo de Engenharia de Software.** Disponível em: <<http://www.labes.ufpa.br/qites/teaching/2006/TopicosES/ProcessoEngenhariaSoftware.pdf>> Acesso em 10 set. 2009.
6. BALZERT, H. **UML 2: compacto.** 1 ed. Rio de Janeiro: Elsevier, 2008. p. 2.
7. Batisti, J. **Artigos SQL Server e VB.NET.** Disponível em: <<http://www.juliobattisti.com.br/>> Acesso em 2009 (durante toda duração do projeto).
8. **Desenvolvimento Iterativo e Incremental.** Disponível em: <http://pt.wikipedia.org/wiki/Desenvolvimento_iterativo_e_incremental> Acesso em 24 set. 2009.
9. Elias, M. **Artigos SQL Server e VB.NET.** Disponível em: <<http://www.linhadecodigo.com.br/>> Acesso em 2009 (durante toda duração do projeto).
10. **IBM Rational Unified Process.** Disponível em: <http://pt.wikipedia.org/wiki/Rational_Unified_Process#Fases> Acesso em 06 set. 2009.

11. **Imagens**. Disponível em: <<http://www.fotosearch.com.br/>> Acesso em 2009 (durante toda duração do projeto).

12. Macoratti, J. C. **Artigos Orientação a Objeto, MVC, SQL Server e VB.NET**. Disponível em: <<http://www.macoratti.net/>> Acesso em 2009 (durante toda duração do projeto).

13. **Microsoft Access**. Disponível em: <http://pt.wikipedia.org/wiki/Microsoft_Access> Acesso em 15 set. 2009.

14. **Microsoft .NET**. Disponível em: <<http://msdn.microsoft.com/pt-br/netframework/default.aspx>> Acesso em 07 set. 2009.

15. **MVC**. Disponível em: <<http://pt.wikipedia.org/wiki/MVC>> Acesso em 09 set. 2009.

16. **O Processo de Software**. Disponível em: <http://www.macoratti.net/proc_sw1.htm> Acesso em 24 set. 2009.

17. PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw-Hill Interamericana do Brasil, 2006. p. 52.

18. REZENDE, D. A. **Engenharia de software e sistemas de informação**. 3 ed. Rio de Janeiro: Brasport, 2005. p. 02.

19. **SQL Server 2005**. Disponível em: <<http://www.microsoft.com/brasil/servidores/sql/2005/default.msp>> Acesso em 08 set. 2009.

20. **UML**. Disponível em: <<http://pt.wikipedia.org/wiki/UML>> Acesso em 08 set. 2009.

21. WAZLAWICK, R. S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 1 ed. São Paulo: Campus, 2004. p. 102.

GLOSSÁRIO

ARQUITETURA DE SOFTWARE: consiste dos componentes de software, suas propriedades externas, e seus relacionamentos com outros softwares.

ARTEFATOS: São os produtos finais ou intermediários produzidos e usados durante o projeto.

BANCO DE DADOS: conjunto de registros dispostos em estrutura regular que possibilita a reorganização dos mesmos e produção de informação.

DATA WAREHOUSE: sistema utilizado para armazenar informações relativas às atividades de uma organização em bancos de dados, de forma consolidada.

DIAGRAMA: representação visual estruturada e simplificada de um conceito.

IMPLEMENTAÇÃO: Implementação é a fase do Ciclo de Vida de um software que corresponde à elaboração e preparação dos módulos necessários à sua execução.

LINGUAGEM DE PROGRAMAÇÃO: É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.

METODOLOGIA: conjunto estruturado de práticas que pode ser repetível durante o processo de produção de software.

MODELAGEM: atividade de construir modelos que expliquem as características ou o comportamento de um software.

PLANO DE GERENCIAMENTO DE PROJETO: contém toda a documentação e os processos requeridos para dirigir, gerenciar, monitorar e controlar o projeto.

PROTÓTIPO: produto que ainda não foi comercializado, mas está em fase de testes ou de planejamento.

SAP: empresa alemã criadora do Software de Gestão de Negócios de mesmo nome.

APÊNDICE 1 – PLANO DE GERENCIAMENTO DO PROJETO (PGP)

APÊNDICE 2 – DIAGRAMA DE CASOS DE USO

APÊNDICE 3 – ESPECIFICAÇÃO DOS CASOS DE USO

APÊNDICE 4 – DIAGRAMA DE CLASSES

APÊNDICE 5 – DIAGRAMAS DE SEQUÊNCIA

APÊNDICE 6 – DIAGRAMA ENTIDADE RELACIONAMENTO

**APÊNDICE 7 – PADRÕES DE NOMENCLATURA DE
DESENVOLVIMENTO**

APÊNDICE 8 – CÓDIGOS FONTE

(Disponibilizado no CD complementar deste projeto)