

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PAULA PIANTINO VITIRITTI

**RUP APLICADO A PROJETOS NA ÁREA DE
AUTOMAÇÃO E CONTROLE**

Monografia apresentada à Escola
Politécnica da Universidade de São
Paulo para conclusão do curso de
MBA em Tecnologia de Software

Área de Concentração:

Engenharia de Software

Orientadora:

Profa. Dra. Jussara Pimenta Matos

São Paulo

2007

MBA/ES
2007
V835n

DEDALUS - Acervo - EPEL



31500020086

FICHA CATALOGRÁFICA

M2007CL

Vitiritti, Paula Piantino

RUP aplicado a projetos na área de automação e controle /
P.P. Vitiritti. -- São Paulo, 2007.
63 p.

Monografia (MBA em Tecnologia de Software) – Escola
Politécnica da Universidade de São Paulo. Programa de Edu-
cação Continuada em Engenharia.

1. Processo unificado automação e controle I. Universidade
de São Paulo. Escola Politécnica. Programa de Educação Conti-
nuada em Engenharia II.t.

1825634

Aos pais, Henriqueta e Felix,
e aos irmãos e amigos
pelo apoio e compreensão.

OK

Agradecimentos

À Profa. Dra. Jussara Pimenta Matos, pela orientação, paciência e o apoio que foram essenciais à execução deste trabalho.

Aos professores e colegas do MBA pela transferência de conhecimento e amizade.

RESUMO

O objetivo deste trabalho é apresentar um processo de desenvolvimento aplicado a um projeto de controle. Este processo foi aplicado em um projeto de controle para uma aplicação em mineração.

Os resultados obtidos estão descritos no estudo de caso. Este projeto inicialmente adotava o processo de desenvolvimento de automação e controle da empresa analisada e apresentava diversos problemas que foram sanados com a adoção do processo mencionado, que melhorou a definição dos requisitos, diminuiu os riscos e colaborou para o cumprimento dos prazos.

A proposta deste processo está inserida em um contexto maior que visa à adaptação do RUP para projetos de diferentes naturezas dentro da empresa, de modo a integrar a metodologia empregada por todos os engenheiros da empresa.

ABSTRACT

The purpose of this essay is to present a development process defined for control projects. It was applied to a control project for an application in mining.

The final results were described in the use case. Initially, this project adopted the process of development of automation and control of the analyzed company and then presented many problems that were solved with the adoption of the process mentioned above, that improved the definition of the requirements, reduced the risks and cooperated with the deadlines.

The proposal of this process is inserted in a bigger context that aims the adaptation of the unified process for projects of different natures in the companies, in order to integrate the used methodology for all the engineers of the company.

SUMÁRIO

1	Introdução.....	1
1.1	Objetivo	1
1.2	Justificativa.....	1
1.3	Estrutura do trabalho	2
2	RUP	3
2.1	Considerações iniciais	3
2.2	RUP – Considerações gerais	3
2.3	Pontos essenciais do RUP	5
2.4	Agilidade com o RUP.....	8
2.5	RUP e XP	9
2.6	Considerações finais.....	13
3	Projetos de automação e Controle.....	14
3.1	Considerações iniciais	14
3.2	Sistemas de controle.....	14
3.3	Norma IEC - 61499	17
3.4	Considerações finais.....	20
4	Processo para projetos na área de automação e controle.....	21
4.1	Considerações iniciais	21
4.2	Processo Proposto.....	21
4.3	Framework.....	23
4.4	Papéis.....	24

4.5	Artefatos	25
4.6	Considerações finais.....	27
5	Estudo de Caso	28
5.1	Considerações iniciais	28
5.2	Projeto Analisado	28
5.3	Processo inicialmente adotado	30
5.3.1	Fluxo de desenvolvimento.....	30
5.3.2	Processo de desenvolvimento.....	33
5.3.3	Problemas do processo	38
5.4	Aplicação da proposta	39
5.4.1	Pontos da metodologia	44
5.4.2	Frameworks criados	46
5.4.3	Papéis.....	46
5.4.4	Artefatos	47
5.4.5	Comparação entre o processo proposto e os processos definidos para a empresa analisada.....	50
5.5	Considerações finais.....	52
6	Conclusão	53
6.1	Trabalhos futuros.....	54
	Referência Bibliográfica.....	55
	Anexo 1: Processo atual de Projetos de Desenvolvimento de Software	57
	Fluxo de desenvolvimento.....	57
	Processo de desenvolvimento.....	60

LISTA DE FIGURAS

Figura 1 – Um ciclo e suas fases e iterações	4
Figura 2 – Principais marcos do processo unificado.....	4
Figura 3 – RUP workflow.	5
Figura 4 – Características dos sistemas DRE.....	15
Figura 5 – Comparação da IEC 61499 e UML no escopo de módulos de função.	19
Figura 6 – Comparação da IEC 61499 e UML no escopo de todo o sistema.	20
Figura 7 – Fluxo de desenvolvimento do Projeto – Etapa Inicial.	31
Figura 8 – Fluxo de desenvolvimento do Projeto – Etapa Intermediária.	32
Figura 9 – Fluxo de desenvolvimento do Projeto – Etapa final.	33
Figura 10 – Ações definidas no projeto e suas dependências.	41
Figura 11 – Fluxo de desenvolvimento do Projeto – Etapa incial.....	58
Figura 12 – Fluxo de desenvolvimento do Projeto – Etapa intermediária.	59
Figura 13 – Fluxo de desenvolvimento do Projeto – Etapa final.	59

LISTA DE TABELAS

Tabela 1 – Mapeamento de artefatos do XP e RUP.....	9
Tabela 2 – Mapeamento de funções e papéis do RUP.....	12
Tabela 3 – Mapeamento de funções e papéis do RUP.....	24
Tabela 4 –Artefatos.....	26
Tabela 5 – Processo do projeto de automação e controle.....	34
Tabela 6 – Etapas das ações.....	42
Tabela 7 – Mapeamento papéis.....	47
Tabela 8 – Artefatos propostos e realizados.....	47
Tabela 9 – Comparaçao entre Artefatos dos processos existentes e proposto.....	50
Tabela 10 – Tabela de responsabilidades, atividades e artefato.....	60

LISTA DE ABREVIATURAS E SIGLAS

CLP	Controlador Lógico Programável
DCS	Distributed Control System
DRE	Distributed Real-time embedded Systems
FB	Function Block
IEC	International Electro-technical Commission
IPMCS	Industrial Process Measurement and Control System
LIMS	Laboratory Information Management System
MTBF	Mean Time between Failure
PDA	Personal Digital Assistant
RTES	Real-time embedded System
RUP	Rational Unified Process
RUP	Rational Unified Process
SCADA	Supervisory Control and Data Acquisition
SDCD	Sistema Digital de Controle Distribuído
UML	Unified Modeling Language
UP	Unified Process
XP	Extremme Programming

1 Introdução

1.1 Objetivo

Atualmente as empresas de projetos atuam em diferentes setores e os projetos têm se mostrado multidisciplinares, de modo que os funcionários destas empresas estão cada vez menos especialistas, podendo atuar em diversas áreas no mesmo projeto. A existência de um processo único que serve de base para os projetos de diferentes áreas torna-se fundamental para tornar a empresa competitiva, na medida em que torna os seus projetos mais ágeis.

O presente trabalho tem como objetivo apresentar uma proposta de adaptação do RUP para projetos de automação e controle. Esta proposta se baseou no RUP, XP e na norma IEC 61499 de desenvolvimento de softwares de controle.

1.2 Justificativa

Este trabalho tem como motivação uma aplicação prática do RUP em uma empresa de médio porte com cerca de 300 funcionários que realiza projetos de engenharia e software para a Indústria Química. A Chemtech possui escritórios no Rio de Janeiro, São Paulo, Belo Horizonte, Salvador, Porto Alegre e Houston. O projeto analisado foi desenvolvido pelo escritório de Belo Horizonte nas instalações do cliente.

A Chemtech desenvolve projetos de automação, controle, simulação, engenharia básica, desenvolvimento de Software, entre outros. De acordo com o tipo de projeto, existe um processo definido diferente. É muito comum que os projetos englobem mais de uma área de atuação, de modo que a equipe se torna responsável por atividades de diferentes áreas de conhecimento.

Desta forma, há a necessidade de definir um processo único para qualquer tipo de projeto. Adotou-se o RUP como base para a definição deste processo. Este trabalho aborda a aplicação do RUP em projetos de automação e controle.

1.3 *Estrutura do trabalho*

O capítulo 2 apresenta algumas características do processo unificado (UP) e suas configurações, além de conceitos de programação extrema (XP) de forma a torná-lo mais ágil.

Posteriormente, no capítulo 3, estão descritas algumas características de sistemas de automação e controle com uma visão geral da norma IEC 61499 (IEC: Comissão Internacional Eletrotécnica) e a sua aplicação com Linguagem de Modelagem Unificada (UML).

No capítulo 4 é feita uma proposta para a utilização do RUP em projetos de automação e controle. Finalmente, é apresentada a aplicação desta proposta em um estudo de caso (capítulo 5).

O capítulo 6 apresenta a conclusão e propostas de trabalhos futuros.

2 RUP

2.1 Considerações iniciais

Este capítulo apresenta inicialmente as principais características do RUP. Posteriormente, são apresentadas formas para deixá-lo mais ágil através da adoção de práticas de XP. O RUP é a base do processo de desenvolvimento de projetos de controle e automação proposto neste trabalho.

2.2 RUP – Considerações gerais

O RUP é um processo de desenvolvimento de software iterativo e incremental criado pela Rational Software Corporation.

De acordo com KRUCHTEN e KROLL, 2003, o processo unificado consiste em um processo de desenvolvimento de Software que é caracterizado por ser iterativo, centrado na arquitetura, dirigido por casos de uso e por riscos.

Cada ciclo do processo unificado é composto por 4 fases: concepção, elaboração, construção e transição (REIS e BELCHIOR, 2003). Cada fase é ainda dividida em iterações, conforme apresentado na Figura 1:

- **Concepção:** fornece uma visão inicial do projeto, com a definição do escopo e a definição preliminar dos riscos;
- **Elaboração:** analisa o domínio do problema, com a proposta da arquitetura da aplicação e suas funcionalidades;
- **Construção:** construção do produto;
- **Transição:** garante a entrega do sistema aos usuários finais. Há a realização de testes e ajustes.

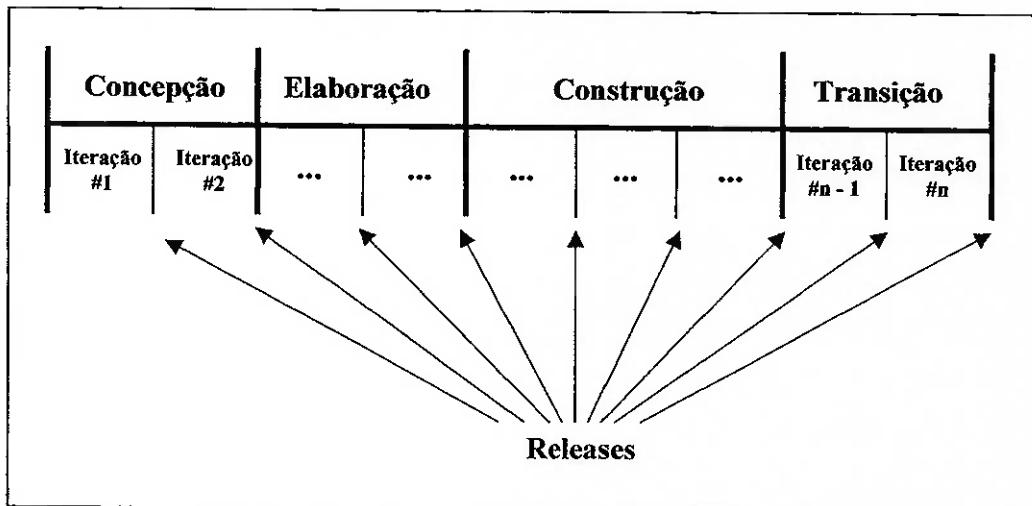


Figura 1 – Um ciclo e suas fases e iterações

(Adaptação de: KRUCHTEN e KROLL, 2003).

O processo unificado possui os marcos principais apresentados na Figura 2 (KRUCHTEN e KROLL, 2003).

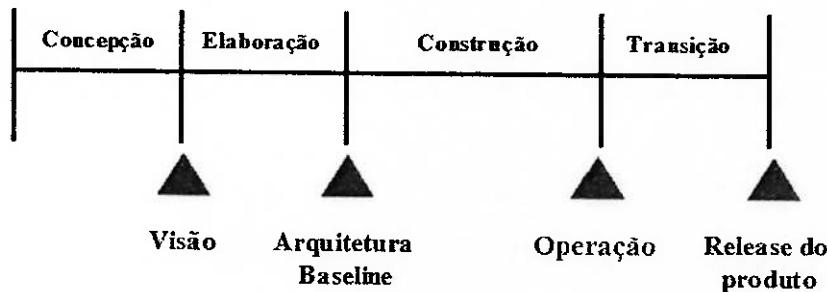


Figura 2 – Principais marcos do processo unificado.

Uma iteração é uma seqüência de atividades definidas em um plano que resultam em uma versão, que possui a característica de ser executável. Durante cada iteração, as atividades são classificadas em 9 disciplinas (Figura 3):

- Disciplinas de engenharia:
 - Modelagem do negócio;
 - Requisitos;
 - Análise e projeto;
 - Implementação;

- Teste;
- Implantação;
- Disciplinas de suporte:
 - Gerência de configuração e mudança;
 - Gerência de projeto;
 - Ambiente.

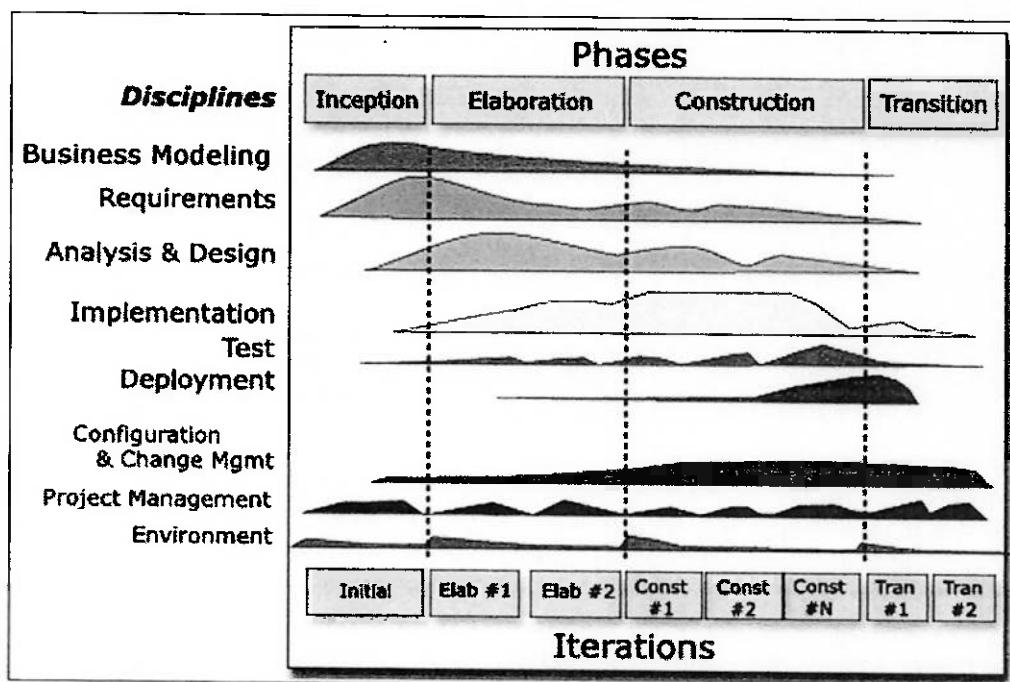


Figura 3 – RUP workflow.

(Fonte: RATIONAL SOFTWARE CORPORATION, 2001)

2.3 Pontos essenciais do RUP

O RUP é um processo que possui como característica ser configurável. De acordo com PROBASCO, 2001, existem 10 pontos do RUP, que independente do tamanho do projeto, devem ser considerados:

PONTO 1: Desenvolver uma visão

De acordo com PROBASCO, 2001, uma visão clara do sistema é a chave para o desenvolvimento de um produto que atenda às necessidades dos clientes. A visão do sistema deve fornecer as seguintes informações:

- Termos principais (glossário);
- Definição do problema;
- Definição dos colaboradores chave, usuários e de suas necessidades;
- Características do produto;
- Requisitos funcionais (casos de uso);
- Requisitos não-funcionais;
- Restrições do projeto.

A visão fornece uma base, muitas vezes contratual, para os requisitos técnicos mais detalhados.

PONTO 2: Gerenciar o plano

No RUP, o plano de desenvolvimento agrega todas as informações necessárias para gerenciar o projeto e pode incorporar outros itens desenvolvidos que devem ser atualizados no decorrer do projeto durante a fase de concepção. No plano de desenvolvimento são apresentados o cronograma do projeto, a definição das iterações, os recursos e ferramentas a serem utilizados.

PONTO 3: Identificação e mitigação dos riscos

Um dos pontos principais do RUP é a identificação e mitigação dos riscos nas fases iniciais do projeto. Para cada risco identificado deve ser desenvolvido um plano para atacá-lo. A partir da identificação dos riscos, são realizados o plano de projeto e a definição das iterações.

PONTO 4: Gerar e controlar versões

A rotulação de versões deve ser feita regularmente de modo a prover um mecanismo de endereçar, comunicar e resolver questões gerenciais, técnicas e riscos de projeto.

PONTO 5: Examinar o caso de negócio

A análise do caso de negócio verifica se o projeto é um investimento válido, ajudando no desenvolvimento do plano econômico do projeto e na análise do ROI.

PONTO 6: Desenvolver a arquitetura dos componentes

O RUP fornece uma forma metodológica e sistemática para projetar, desenvolver e validar a arquitetura do sistema. Definem-se uma arquitetura candidata, o refinamento da arquitetura, a análise do comportamento, o projeto dos componentes do sistema, entre outros.

O RUP apresenta a arquitetura através de um documento que descreve as diferentes visões da arquitetura, que endereçam diferentes usuários e stakeholders. A arquitetura apresenta a estrutura do sistema e a interação entre os componentes.

PONTO 7: Construir e testar o produto incrementalmente

Uma das bases do RUP é o desenvolvimento incremental, ou seja, incrementalmente codificar, construir e testar os componentes do sistema através do ciclo de vida do projeto, produzindo versões executáveis no final de cada iteração.

PONTO 8: Verificar e avaliar os resultados

No final de cada iteração do RUP é necessário avaliar os resultados para verificar a aderência aos resultados esperados, tanto no nível do produto quanto do processo. Para tanto, faz-se necessária uma revisão formal dos testes.

PONTO 9: Gerenciar e controlar modificações

No processo do RUP, um dos pontos principais é o gerenciamento e controle do escopo do projeto através das modificações que ocorrem ao longo do ciclo de vida do

projeto. O objetivo é considerar todas as necessidades dos stakeholders e procurar alcançá-las sempre que possível, ao mesmo tempo que um produto de qualidade é entregue dentro do prazo.

Ao longo do desenvolvimento do sistema, os stakeholders vão requisitar mudanças que muitas vezes são incorporadas, devendo assim ser gerenciadas através de um processo consistente.

PONTO 10: Dar suporte ao usuário

De acordo com o RUP, todo o produto deve ser entregue com todos os materiais necessários para auxiliar o usuário final a usar e manter o produto. Desta forma, deve ser produzido, pelo menos, o manual do usuário, o manual de instalação e os releases notes, além de uma lista com todos os documentos entregues com o produto.

2.4 Agilidade com o RUP

Segundo KRUCHTEN, 2001, agilidade é a capacidade de adaptar e de reagir apropriadamente às mudanças do ambiente e às demandas impostas por este ambiente. Um processo ágil não está somente ligado ao tamanho do processo ou à rapidez da entrega, mas sim em sua flexibilidade.

O RUP é um processo que deve ser configurado de acordo com o projeto, uma forma de tornar o processo mais ágil, segundo KRUCHTEN, é prover meios de ajudar o gerente do projeto a fazer um conjunto de decisões certas para o caso de desenvolvimento inicial. Para tanto, ele propõe quatro formas de atacar esta questão:

- **Configurações pré-definidas:** prover instâncias pré-definidas do framework do RUP, nas quais algumas decisões já foram feitas para um dado ambiente de desenvolvimento de Software;
- **RUP Componentizado:** organizar o framework do RUP para manipular pequenas áreas de conhecimento do processo, através da componentização do RUP;

- **Ferramentas para a configuração do RUP:** fornecer ferramentas de suporte para a construção da configuração do RUP;
- **Ferramentas para criação de processos dos usuários:** fornecer ferramentas de suporte para permitir aos usuários expandir o framework do RUP e criar os seus próprios componentes de processo.

2.5 RUP e XP

O XP (programação extrema) é uma metodologia de desenvolvimento ágil que provê flexibilidade, rapidez e alta qualidade (BECK, 2000). O XP prioriza as funcionalidades que representam maior valor agregado ao projeto. Possui um conjunto de valores, princípios e práticas que são bem diferentes das formas tradicionais de desenvolvimento de Software. Atualmente a competição entre as empresas está cada vez mais acirrada, de modo que a adoção de medidas que podem deixar o seu processo de desenvolvimento mais ágil é muito importante para a sobrevivência da empresa.

SMITH, 2001 comparou o RUP e o XP, considerando os seus artefatos, papéis, atividades, disciplinas e fluxos de trabalho, com os seguintes resultados principais:

Artefatos

A Tabela 1 apresenta o mapeamento dos artefatos do RUP e XP para projetos de ciclo curto.

Tabela 1 – Mapeamento de artefatos do XP e RUP.

Artefato do XP	Artefato do RUP
Estórias	Visão
Documentações adicionais de conversas	Glossário

Artefato do XP	Artefato do RUP
	Modelo de casos de uso
Restrições	Especificações complementares
Testes de aceitação Testes unitários Dados de teste Resultados de teste	Modelo de teste
Código do Software	Modelo de implementação
Releases	Produto Release notes
Metaphor	Documento de arquitetura do Software
Design – CRC, UML, esquemas Tarefas Tarefas técnicas Documentos de design – produzidos no final do projeto Documentação de apoio	Modelo do design
Padrões de codificação	Guias de design Guias de programação
Ambiente de trabalho Ferramentas de teste	Ferramentas
Plano de release Estimativas das estórias Estimativas das tarefas Plano de iteração	Plano de desenvolvimento de Software Plano de iteração

Artefato do XP	Artefato do RUP
Plano geral – orçamento	Caso de negócio Lista de risco Plano de aceitação do produto
Relatórios sobre o progresso das tarefas Registro do tempo de execução das tarefas Dados medidos: recursos, escopo, qualidade, tempo Outras medidas Relatórios e anotações de reuniões	Avaliação do status do projeto
Defeitos e dados associados	Requisições de mudanças
Ferramentas de gerenciamento de código	Plano de gerenciamento de configurações Repositório do projeto Workspace
Spike	Protótipos
	Caso de desenvolvimento Templates específicos do projeto

Atividades

As atividades do RUP são bem definidas e mostram como algo deve ser feito. Já as atividades do XP são informais e consistem no trabalho que deve ser realizado por um papel utilizando artefatos de entrada e produzindo ou modificando artefatos.

Disciplinas e fluxo de trabalho

As seguintes disciplinas do XP poderiam ser implementadas no RUP segundo Smith. Algumas delas já estão incorporadas no RUP.

- Programação em pares;
- Padrões de codificação;
- Integração contínua.

Papéis

A Tabela 2 apresenta o mapeamento dos papéis do RUP.

Tabela 2 – Mapeamento de funções e papéis do RUP.

Função	Papel no RUP
Gerente de Projetos	Gerente de projetos Engenheiro de processos Gerente de deployment Revisor de requisitos Revisor de arquitetura
Executivo	Revisor do projeto Stakeholder Revisor de requisitos
Programador chefe	Analista de sistemas Especificador de requisitos Projetista da interface com o usuário Arquiteto de Software Revisor do design Engenheiro de processos

Função	Papel no RUP
	Especialista das ferramentas Gerente de configuração Gerente do controle de mudanças
Programador	Projetista Implementador Revisor de Código Integrador Projetista de testes Testador
Assistente administrativo	Auxilia o gerente de projetos no planejamento de atividades e cronograma Auxilia o gerente de controle de mudanças a controlar as mudanças nos artefatos Auxilia demais papéis quando necessário

2.6 Considerações finais

O RUP é um processo de desenvolvimento de software que possui como característica ser iterativo e incremental. O RUP pode ser configurado de acordo com o tipo do projeto e suas necessidades.

Existem diversas propostas de configuração do RUP para transformá-lo mais ágil. Um paralelo entre o RUP e o XP (metodologia de desenvolvimento ágil) pode ser estabelecido, definido um novo processo que engloba características dos dois processos.

O RUP será a base do processo proposto neste trabalho e também serão adotadas algumas práticas de XP de modo a deixá-lo mais simples e ágil.

3 Projetos de automação e Controle

3.1 Considerações iniciais

Os sistemas de controle industrial são tipos de sistemas embarcados, de tempo real e distribuídos (DRE). Eles são utilizados no controle regulatório e de segurança das plantas e foram abordados no início deste capítulo.

Linguagens, metodologias e padrões de desenvolvimento de sistemas de controle estão definidos na norma IEC 61499, conforme apresentado no item 3.3. Este item também enumera alguns pontos fracos desta norma, que podem ser cobertos através de conceitos de orientação a objetos e UML.

O conhecimento das características dos sistemas de controle e metodologias e padrões existentes para a sua definição é essencial para a especificação do processo de desenvolvimento de sistemas de controle proposto neste trabalho.

3.2 Sistemas de controle

Os sistemas embarcados são sistemas de computadores especializados que incluem software e hardware. São desenvolvidos para tipos específicos de aplicações. Estão presentes em máquinas industriais, automóveis, equipamentos médicos, câmeras, aparelhos domésticos, aviões, telefones celulares e computadores pessoais portáteis (PDAs). Estes sistemas têm recursos limitados de hardware e software.

Os sistemas embarcados de distribuídos de tempo real (DRE) são um tipo de sistemas embarcados que apresentam as funções mostradas na Figura 4, segundo GONG, 2003.

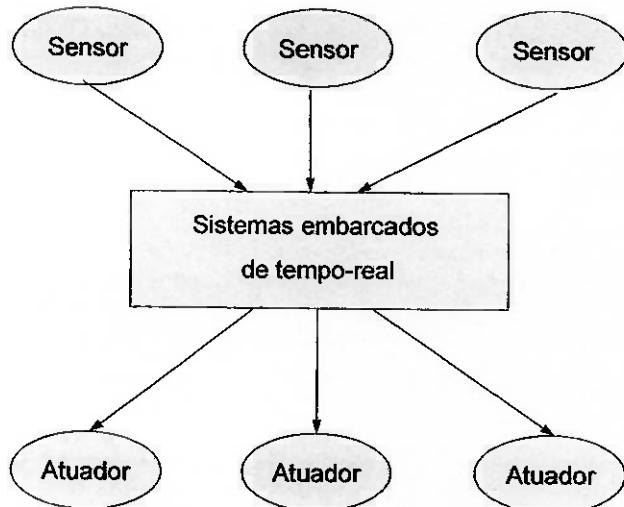


Figura 4 – Características dos sistemas DRE.

- **Leitura:** O sistema DRE lê os dados dos sensores de entrada e então os processam. Os sensores possuem informações sobre o processo, como temperatura, pressão, posição, status, entre outros.
- **Controle:** O sistema DRE gera comandos sobre os atuadores que estão associados a instrumentos do campo, como motores e válvulas, por exemplo.

Segundo KRISHNA *et al.*, 2001, os sistemas DRE ainda possuem limitações de memória, eles devem suportar diferentes linguagens de programação, assim como diferentes sistemas operacionais e tipos de processadores. Os sistemas de controle de aplicações industriais são exemplos de sistemas DRE.

De acordo com HECK *et al.*, 2003, de 60 a 80% do desenvolvimento de um sistema de controle complexo é o esforço para o desenvolvimento do Software, enquanto de 20 a 40% é o tempo para projetar o sistema de controle, ou seja, para elaborar a lógica de controle.

Nos sistemas de controle são desenvolvidas lógicas de controle e de intertravamento, que agem sobre os atuadores:

- **Lógicas de intertravamento de segurança e alarmes**, que garantem a segurança de instalações industriais através da atuação em elementos do

campo (ligando ou desligando equipamentos, por exemplo) quando as variáveis do processo estão saindo da faixa especificada;

- **Lógicas de controle regulatório**, que controlam as variáveis do processo mantendo-as próximas dos valores especificados, através da atuação em válvulas de controle e motores, por exemplo.

O CLP (controlador lógico programável) e o SDCD (sistema digital de controle distribuído) são sistemas de controle de aplicações industriais, possuindo as características de serem embarcados e de tempo-real.

Historicamente, os CLPs eram utilizados para o processamento de lógicas de intertravamento de segurança e alarmes, enquanto que os SDCDs eram utilizados no controle regulatório. Porém, com o desenvolvimento da tecnologia, ambos os sistemas possuem as mesmas atribuições e a separação entre estes sistemas é realizada de acordo com a sua aplicação, envolvendo custos e facilidade de implementação e manutenção.

Existem diversas linguagens de programação dos CLPs e SDCDs, como Ladder e FBD. Além disso, também existem diversos protocolos de comunicação entre o CLP ou SDCD e o campo (sensores e atuadores), como Modbus e OPC.

Além do controle executado automaticamente pelos sistemas de controle, faz-se necessária a interação humana com estes sistemas para o completo controle da planta. Esta interação é realizada para definir qual é o valor desejado de uma variável, ligar ou desligar um equipamento, por exemplo. Além disso, os operadores devem ter conhecimento das condições do processo para poderem tomar estas ações, além de garantirem a segurança da planta. Estas atividades são realizadas através dos sistemas supervisórios que permitem a monitoração das variáveis do processo e envios de comandos para o campo. Os sistemas supervisórios possuem interface gráfica para permitir esta interação. Eles também são chamados de SCADA (Supervisory Control and Data Aquisition).

3.3 Norma IEC - 61499

A norma IEC 61499 define conceitos básicos e a metodologia a ser utilizada por projetistas de sistemas para produzir sistemas modulares e interoperáveis para aplicações na indústria em controle (TRANORIS *et al.*, 2003). Foi designada para os sistemas de controle com foco nos SDCDs, baseia-se na norma anterior a IEC 61131 definida para os CLPs (PANJAITAN *et al.*, 2006).

A comissão internacional eletrotécnica (IEC) definiu que os blocos de função são construções básicas para o desenvolvimento de aplicações de controle reutilizáveis, interoperáveis e distribuídas. Os blocos de função não exploram os avanços recentes da engenharia de software (THRAMBOULIDIS, 2004).

O bloco de função é uma unidade funcional de Software constituída por uma estrutura de dados individual que depende do tipo de bloco, cujos dados persistem de uma invocação até a próxima. A funcionalidade do bloco de função é definida por meio de algoritmos que processam dados de entrada, dados internos e geram dados de saída (THRAMBOULIDIS, 2001).

Nesta norma, define-se o modelo de tarefas de engenharia a serem realizadas durante as fases de design, implementação, operação e manutenção de sistemas de controle distribuídos. Especifica requisitos genéricos para ferramentas de software para darem apoio a tarefas de engenharias (THRAMBOULIDIS, 2003).

A IEC 61499 possui conceitos bem definidos para a construção e implementação de DCS, mas ela não dá suporte a outras etapas do processo de desenvolvimento, por exemplo, para as etapas iniciais:

- Análise de requisitos;
- Decisões iniciais de alto nível sobre o design;
- Análise da estrutura do sistema e funcionalidades.

Atualmente a indústria de desenvolvimento de softwares industriais tem sofrido uma grande pressão para o desenvolvimento de softwares cada vez mais complexos, dentro de tempos e orçamentos menores. Tem-se a necessidade de um novo método

de desenvolvimento que lide com todas as fases de desenvolvimento: análise, modelagem, construção, implementação, validação, além de ferramentas para dar suporte a estas fases (WEHRMEISTER *et al.*, 2005 e PANJAITAN *et al.*, 2006).

O conceito de bloco de função definido pela IEC 61499 é do tipo procedural, não explorando os benefícios trazidos pela Orientação a Objetos, que aumenta a eficiência do processo de desenvolvimento de Softwares.

A utilização da orientação a objetos no desenvolvimento de software industriais tem crescido bastante, devido a suas características, tais como: abstração de dados, encapsulamento, polimorfismo, concorrência, reusabilidade e interoperabilidade. Além disso, a orientação a objetos produz modelos que são mais fáceis de serem entendidos pelos usuários finais. A utilização de orientação a objetos produz sistemas que são mais fáceis de serem modificados, expandidos e mantidos (THRAMBOULIDIS, 2001).

É possível considerar a utilização do UML no desenvolvimento de sistemas de controle, uma vez que descreve o sistema considerando todo o processo de desenvolvimento. Os diagramas UML, como diagramas de classe, pacote, seqüência, casos de uso e estados podem ser transformados automaticamente em modelos definidos pela IEC 61499 (PANJAITAN *et al.*, 2006 e LEE e SHIN, 2006).

Foram propostos trabalhos que relacionam os conceitos de blocos de função e UML com os seguintes focos (PANJAITAN *et al.*, 2006, THRAMBOULIDIS, 2003 e LEE e SHIN, 2006):

- Modelar todo o sistema em UML e depois converter para o ambiente FB;
- Utilizar o UML somente nos pontos em que a IEC 61499 não cobre.

A Figura 5 e a Figura 6 comparam os blocos de função e o UML, com duas abordagens: módulos de função e sistema como um todo, respectivamente.

O comportamento do modelo da IEC 61499 é limitado ao nível do bloco de função (Figura 5), a máquina de estados ECC se aplica somente aos blocos de função básicos; o diagrama de seqüência temporal é uma ferramenta de modelagem utilizada

somente para descrever a seqüência de comunicação entre outros blocos de função. Por outro lado, os diagramas UML apresentam em maiores detalhes o comportamento dentro dos algoritmos ligados aos estados (por exemplo, através de diagramas de atividade), associados aos estados de um diagrama de estados usados com o mesmo propósito de um ECC na IEC 61499. Assim, uma descrição lógica dos algoritmos pode ser modelada no UML com maiores detalhes.

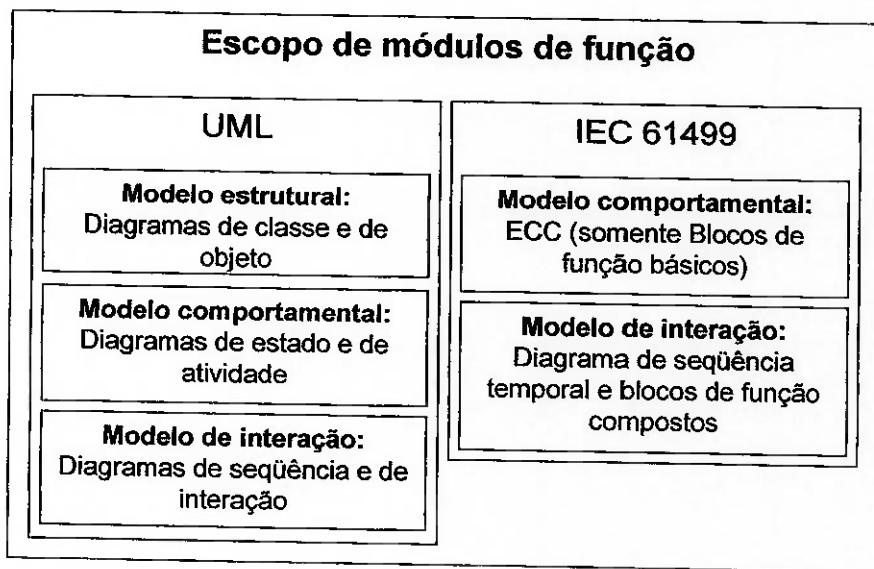


Figura 5 – Comparação da IEC 61499 e UML no escopo de módulos de função.

(Fonte: PANJAITAN e FREY, 2006)

A IEC 61499 fornece alguns modelos de referência para a construção de um sistema, a interação entre os modelos como interconexões entre os blocos de função pode ser observada neste modelo. No nível de implementação, os parâmetros de execução e os protocolos de rede do sistema podem ser configurados. Este modelo pode ser enriquecido por uma magnitude de diagramas em UML para apresentar ao desenvolvedor os requisitos que caracterizam alguns cenários que podem ser apresentados através de diagramas de atividade que descrevem o fluxo do processo no nível do sistema. A interação entre os componentes pode ainda ser modelada usando diagramas de seqüência e/ou diagramas de interação (Figura 6).

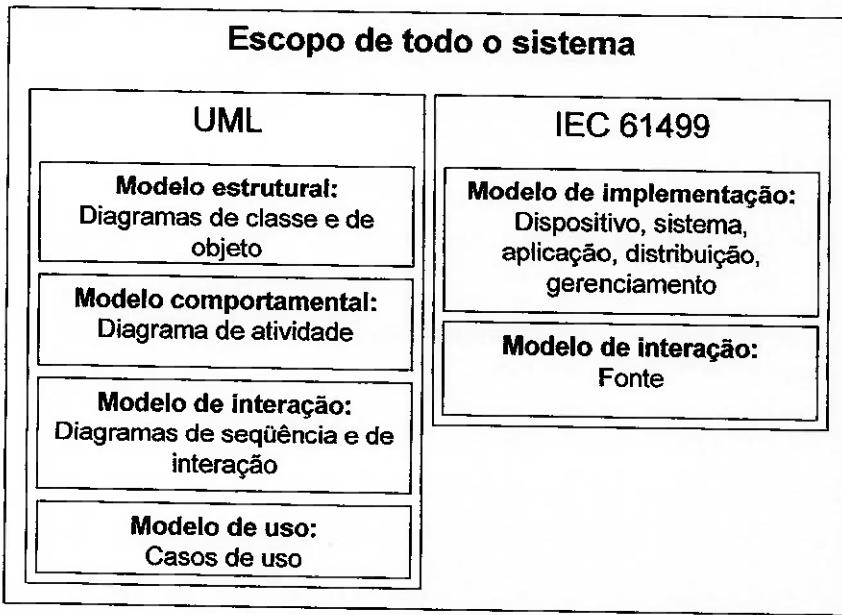


Figura 6 – Comparação da IEC 61499 e UML no escopo de todo o sistema.

(Fonte: PANJAITAN e FREY, 2006)

3.4 Considerações finais

Os sistemas de controle estão cada vez mais complexos e a disputa entre as empresas mais acirrada, de modo que a existência de um processo de desenvolvimento e configuração mais ágil é crucial.

A norma IEC 61499 é específica para o desenvolvimento de sistemas de controle, porém ela possui algumas deficiências quanto à definição de requisitos, arquitetura e funcionalidades. Além de não considerar o processo como um todo.

A fim de suprir estas deficiências, neste trabalho é proposta a adoção do RUP em paralelo à norma IEC 61499. O RUP será utilizado na medida em que considera todo o processo de desenvolvimento e possui artefatos bem definidos que podem ser construídos através da utilização de diagramas UML, os quais serão responsáveis por modelar o sistema e auxiliarão na construção dos blocos de função definidos pela norma IEC 61499. Da norma mencionada, serão mantidos os padrões de desenvolvimento e linguagem.

4 Processo para projetos na área de automação e controle

4.1 Considerações iniciais

Este capítulo apresenta a proposta de adaptação do RUP para projetos de automação e controle. Inicialmente estão descritos quais os pontos da metodologia RUP que serão empregados neste processo, além de práticas para tornar este processo mais ágil.

Este processo visa suprir as deficiências da norma IEC 61499 e propõe a utilização de diagramas UML que auxiliam a construção dos blocos de função definidos pela norma mencionada.

4.2 Processo Proposto

O processo proposto incorpora o conceito de desenvolvimento incremental e iterativo do RUP e absorveu também as suas fases e disciplinas (item 2.2).

Foram adotados os 10 pontos básicos da metodologia do RUP que são considerados essenciais por PROBASCO, 2001, conforme apresentado no item 2.3. Os pontos 7 e 8 foram adaptados à realidade dos projetos de automação e controle.

- **PONTO 1: Desenvolver uma visão;**
- **PONTO 2: Gerenciar o plano;**
- **PONTO 3: Identificar e mitigar os riscos;**
- **PONTO 4: Gerar e controlar versões;**
- **PONTO 5: Examinar o caso de negócio;**
- **PONTO 6: Desenvolver a arquitetura dos componentes;**

- **PONTO 7: Construir e testar o produto incrementalmente:**
 - Para os projetos de lógica: construir blocos de lógicas e implementá-los incrementalmente;
 - Para os projetos de interface: dividir as telas de acordo com a sua posição na estrutura hierárquica e desenvolve-las incrementalmente.
- **PONTO 8: Verificar e avaliar os resultados:**
 - Para os projetos de lógica de controle e intertravamento:
 - Testar a comunicação com o campo;
 - Realizar o teste do ponto a ponto com o campo;
 - Testar as lógicas simulando entradas provenientes do campo e observar o comportamento.
 - Para os projetos de interface gráfica:
 - Testar as ligações, amplitudes, animações de todos os componentes da interface.
 - Validar o cadastro dos pontos na base de dados
- **PONTO 9: Gerenciar e controlar modificações;**
- **PONTO 10: Dar suporte ao usuário.**

A fim de tornar o processo mais ágil, foram adotados do processo XP:

- Programação em pares;
- Padrões de codificação;
- Integração contínua.

O processo descrito neste trabalho possui as características básicas do RUP, além de adotar alguns pontos do XP, porém possui algumas particularidades que foram

definidas para adequá-lo à realidade e necessidade dos projetos de automação e controle. Os itens a seguir descrevem os seus artefatos, papéis e framework criados.

4.3 Framework

Utilizando a proposta de Kruchten (item 2.4), este processo prevê a criação de um framework de componentes principais pré-definidos de acordo com o tipo de projeto, de modo a deixar o processo mais ágil e pré-configurado:

- Projetos de interface gráfica:
 - Componentes:
 - Bombas;
 - Válvulas;
 - Colunas;
 - Reservatórios;
 - Outros.
 - Alarmes;
 - Base de dados;
 - Comunicação entre o sistema de controle e o campo (sensores e atuadores), através dos protocolos:
 - OPC;
 - Modbus.
- Projetos de lógicas de controle e intertravamento:
 - Sistema:
 - Projetos em CLPs;
 - Projetos em SDCDs;
 - Projetos em sistemas especialistas.

- Funcionalidade:
 - Lógica de intertravamento;
 - Lógica de controle.
- Linguagem de programação:
 - Ladder;
 - FBD.

4.4 Papéis

Neste processo, fez-se o seguinte mapeamento dos papéis, considerando os papéis do RUP e do XP (item 2.5):

Tabela 3 – Mapeamento de funções e papéis do RUP.

Processo Proposto	Papel no RUP
Gerente de Projetos	Gerente de projetos
Líder de Projeto	Engenheiro de processos Revisor de requisitos Revisor de arquitetura Gerente de configuração Gerente do controle de mudanças
Arquiteto do Sistema	Arquiteto de Software
Projetista da Lógica	Especificador de requisitos
Projetista da Interface	Especificador de requisitos Projetista da interface com o usuário Projetista
Programador	Implementador Revisor de Código

Processo Proposto	Papel no RUP
	Integrador
Testador	Projetista de testes Testador
Assistente administrativo	Auxilia o gerente de projetos no planejamento de atividades e cronograma Auxilia o gerente de controle de mudanças a controlar as mudanças nos artefatos Auxilia demais papéis quando necessário

Neste processo, faz-se necessária a criação do papel do projetista da lógica, que possui como atribuição a definição dos requisitos e elaboração da estratégia de controle a ser adotada. Este papel é de responsabilidade do engenheiro de controle e automação.

4.5 Artefatos

Os artefatos a serem construídos neste processo são definidos com base nos artefatos do RUP, do XP e do mapeamento entre os artefatos da norma IEC 61499, conforme apresentado na Tabela 4.

O documento de visão é utilizado para auxiliar no levantamento dos requisitos do sistema, que é um ponto fraco da metodologia prevista pela norma IEC 61499.

Toda a codificação segue a metodologia proposta pela norma IEC 61499, porém para auxiliar a construção dos blocos de função, serão construídos os diagramas de estado, atividade e seqüência.

Tabela 4 –Artefatos.

Artefato da Proposta	Artefato RUP	Obrigatório
Especificação funcional do sistema	Documento de visão	Sim
Plano de Projeto	Plano de desenvolvimento de Software Plano de iteração Lista de risco Caso do negócio Plano de gerenciamento de configurações	Sim
Casos de uso	Casos de uso	Não
Protótipo	Protótipo	Não
Diagrama de estado	Diagrama de estado	Não
Diagrama de seqüência	Diagrama de seqüência	Não
Diagrama de atividade	Diagrama de atividade	Não
Documento de arquitetura	Documento de arquitetura do Software	Não
Checklist do teste unitário	Modelo de teste	Não
Plano de testes e casos de teste		Não
Relatório de teste		Sim
Manual de Operação; Manual de instalação e Configuração	-	Sim

Artefato da Proposta	Artefato RUP	Obrigatório
Documento com o aceite do cliente	Plano de aceitação do produto	Sim

4.6 Considerações finais

O RUP, como um processo composto por fases e disciplinas, foi incorporado ao processo proposto. Porém, por se tratar de um processo configurável, foram adotados alguns pontos chave de sua metodologia, ao quais foram adaptados para a realidade dos projetos de automação e controle.

Maior agilidade e flexibilidade a este processo puderam ser alcançadas ao serem empregados frameworks de componentes, de acordo com a natureza do projeto. Além disso, adotou-se a metodologia da programação em pares proposta pelo XP.

Os papéis e artefatos foram definidos a partir dos papéis e artefatos do RUP e do XP. Além destes papéis, foi introduzido o papel do projetista da lógica de controle que deverá ser desempenhado pelo engenheiro de controle para a especificação das lógicas de controle.

Os artefatos a serem produzidos foram classificados em obrigatórios e não-obrigatórios. Grande parte deles é oriunda do RUP e visam suprir as deficiências da metodologia proposta pela norma IEC 61499. Os padrões de desenvolvimento e linguagens seguem a proposta da norma supramencionada.

5 Estudo de Caso

5.1 Considerações iniciais

Este capítulo apresenta o estudo de caso de aplicação da proposta de processo descrita no item 4.

Inicialmente é apresentado o projeto abordado no estudo de caso. A seguir, está descrito o processo de desenvolvimento de projetos de automação e controle atualmente adotado pela empresa responsável pelo projeto apresentado neste estudo de caso. Neste tópico também estão abordados os problemas deste processo.

Finalmente, são apresentados os resultados da adoção da metodologia proposta, onde é realizada uma comparação entre o processo proposto e os processos de automação e controle e de desenvolvimento de Software da empresa.

5.2 Projeto Analisado

O estudo de caso corresponde a um projeto de automação e controle realizado durante 8 meses para uma empresa de mineração.

Todo o projeto foi desenvolvido nas instalações do cliente, envolvendo duas grandes frentes de trabalho:

1. Diagnóstico da instrumentação:

- a. Análise da aferição dos instrumentos;
- b. Proposta de melhorias de formas de medição (proposta de novos tipos de instrumentos de medição);
- c. Análise de criticidade de falhas na instrumentação e propostas de redundância.

2. Desenvolvimento de novas lógicas de controle:

- a. Análise das lógicas existentes;
- b. Proposta de novas lógicas de curto e longo prazo;
- c. Execução das lógicas de curto prazo.

Este projeto está em andamento e está sendo realizado pela seguinte equipe:

- Um gerente de projetos;
- Um líder de projetos;
- Um consultor;
- Um engenheiro sênior;
- Dois engenheiros plenos;
- Quatro engenheiros júnior.

Neste projeto estão envolvidos os clientes das áreas de operação, manutenção e automação da planta.

O projeto inicialmente adotou o processo de desenvolvimento de sistemas de automação e controle existente na empresa, porém foram observados vários problemas que tornaram este processo inapropriado, de modo que foi adotado o processo descrito no item 4.

Este estudo de caso abrange a frente de trabalho definida no item 2 (desenvolvimento de novas lógicas controle). A frente de trabalho 1 (diagnóstico da instrumentação) não está descrita neste estudo de caso, pois foi concluída antes do início do emprego do processo proposto neste trabalho.

5.3 Processo inicialmente adotado

No início do projeto descrito neste estudo de caso, adotou-se o processo de desenvolvimento de projetos de automação e controle definidos pela Chemtech. Este processo foi substituído pelo processo proposto no item 4.

Os itens a seguir descrevem o fluxo e o processo de desenvolvimento definidos pela empresa e, posteriormente, estão apresentados os problemas deste processo.

5.3.1 Fluxo de desenvolvimento

As figuras 7, 8 e 9 apresentam o fluxo de desenvolvimento do projeto definido as atividades atribuídas ao gerente ou ao líder do projeto e a equipe de projeto. O fluxo apresentado é geralmente utilizado na maior parte dos projetos da empresa. Às vezes, as responsabilidades atribuídas ao líder de projeto, são na verdade, desenvolvidas pelo Gerente do Projeto.

Inicialmente é realizada uma reunião com o Gerente Comercial e com o Líder/Gerente do Projeto, na qual é apresentado o escopo do projeto, o prazo, custos, lucro estimado, restrições, entre outros. Após esta reunião é elaborado o plano do projeto, usando como ferramenta o MS Project, além do WBS. Após esta etapa, utilizando uma ferramenta da Intranet da Empresa, é definido o início do projeto, conforme apresentado na Figura 7.

O Líder do projeto faz o levantamento de dados. Após esta etapa, é realizada uma reunião com o cliente para a validação do escopo e do Plano do Projeto. Em seguida, é realizada uma reunião com a equipe do projeto, na qual é apresentado o sistema a ser desenvolvido, o plano do projeto e as atribuições de cada membro da equipe.

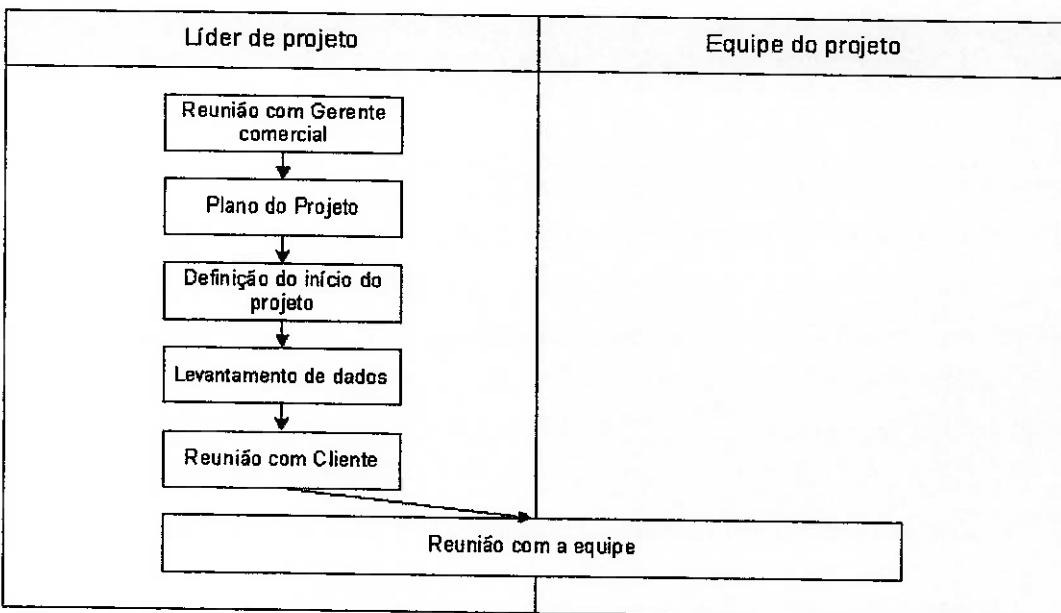


Figura 7 – Fluxo de desenvolvimento do Projeto – Etapa Inicial.

Na etapa intermediária do fluxo de desenvolvimento (Figura 8), o líder de projeto elabora a definição da arquitetura do sistema, realiza os testes de comunicação para a sua validação e elabora a especificação técnica, o plano e a especificação de testes. A equipe de projeto prepara o ambiente de desenvolvimento, desenvolve um protótipo e inicia o desenvolvimento do sistema, sempre acompanhado por testes unitários.

No final desta fase, são gerados os manuais de instalação, operação e configuração, além de serem realizados testes integrados no ambiente de desenvolvimento. Após os testes no ambiente de desenvolvimento, realizam-se testes no ambiente de testes do cliente. Todos os testes são realizados de acordo com o plano de testes.

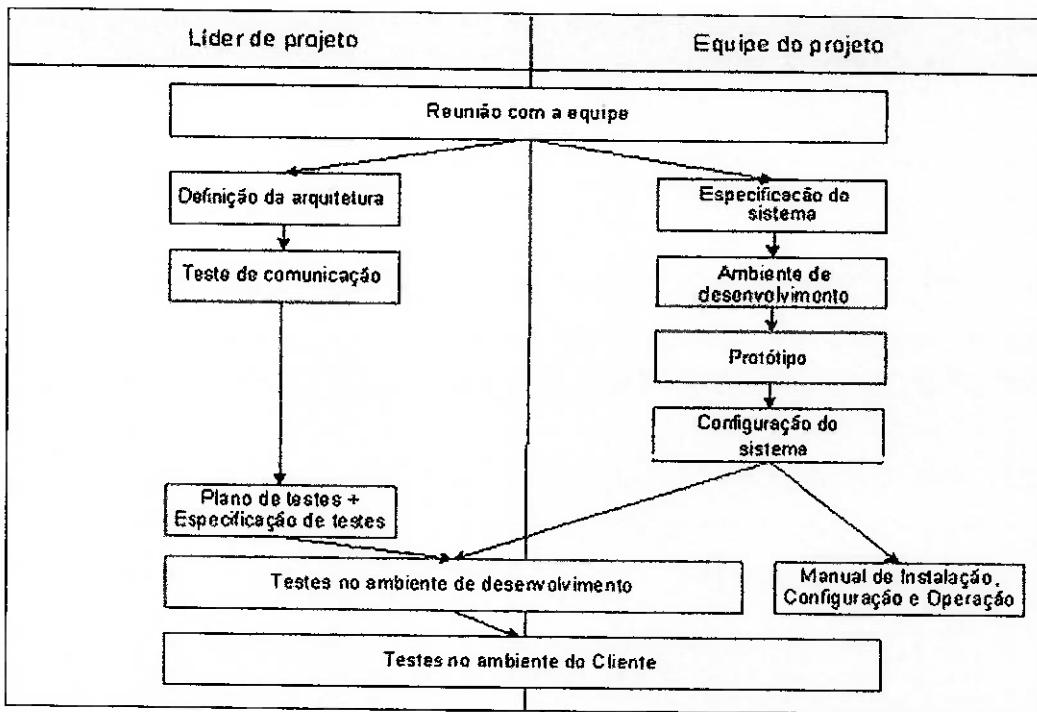


Figura 8 – Fluxo de desenvolvimento do Projeto – Etapa Intermediária.

Uma vez validado com o cliente, são gerados os documentos com os resultados dos testes e o sistema é implantado no ambiente de produção e inicia um período de operação assistida. O projeto então recebe o aceite do cliente. Uma avaliação do projeto é completada pelo cliente e o projeto é fechado na Intranet da Chemtech (Figura 9).

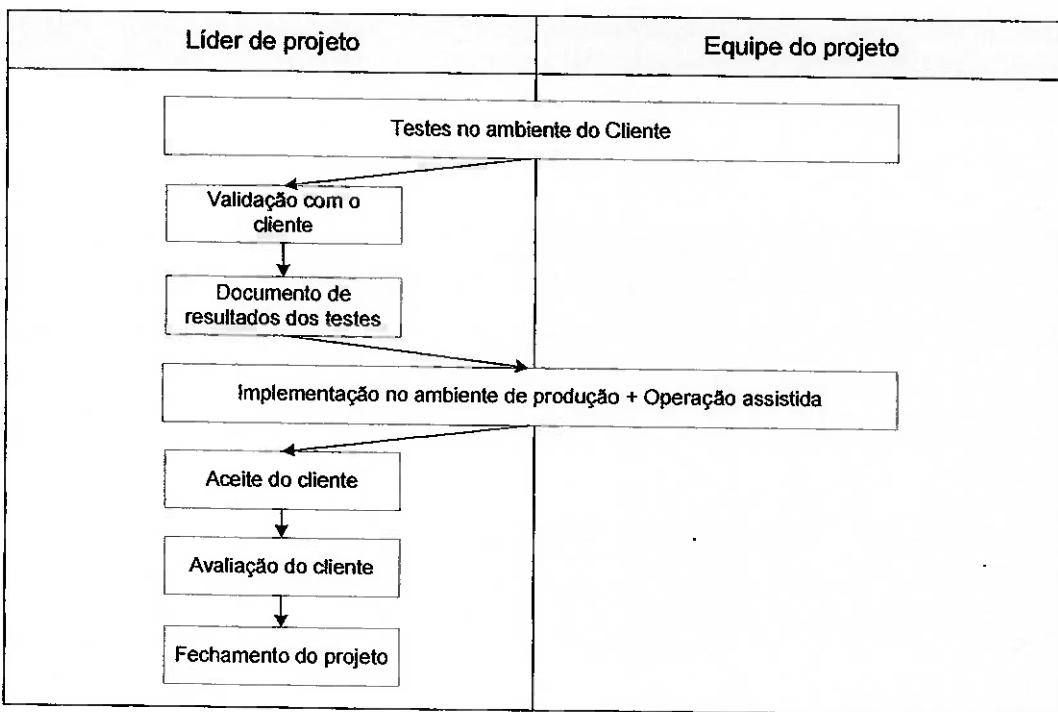


Figura 9 – Fluxo de desenvolvimento do Projeto – Etapa final.

De acordo com este processo, os diversos papéis são divididos entre o líder/gerente de projetos e a equipe do projeto. O líder do projeto é responsável por definições da arquitetura do sistema e validações, já a equipe de projeto tem como atribuições principais a especificação, desenvolvimento e implementação do sistema.

5.3.2 Processo de desenvolvimento

Os projetos de automação e controle da Chemtech foram certificados pela ISO-9001, possuindo o seu processo de desenvolvimento definido pelo documento PO-CHT-AUTO-001 (Procedimento Operacional – Projetos de Automação Industrial).

Os projetos de automação e controle podem envolver duas frentes de trabalho:

1. Desenvolvimento da Interface Gráfica (por exemplo, sistema supervisório);
2. Desenvolvimento da lógica de controle regulatório e de intertravamento e segurança (CLP, por exemplo).

A Tabela 5 apresenta as etapas envolvidas no projeto de automação e controle.

Tabela 5 – Processo do projeto de automação e controle.

Etapa	Descrição	Responsável	Registros Obrigatórios
Procedimentos Iniciais	Preparação do Projeto: estudo preliminar das principais unidades do cliente;	Gerente do projeto; Líder do projeto ou engenheiro responsável.	Definição da metodologia do projeto, consolidação do escopo e definição do cronograma.
Definição da Arquitetura	Reunião de abertura do projeto: apresentação da equipe, da metodologia do projeto, do escopo do projeto, do cronograma.		
Testes de comunicação	Definição do sistema de controle e da arquitetura (SCADA ou SDCD); quantitativo de entradas e saídas.	Equipe do projeto	Especificação da Arquitetura, de Software e Hardware e da Lista de Entradas e Saídas.
Especificação do Sistema	Validação da solução especificada, através da montagem do ambiente de desenvolvimento, configurando um ponto de leitura e um de escrita, verificação da comunicação.	Equipe do Projeto	Realização de testes em bancada
	Definição de:	Equipe do Projeto	Especificação Funcional
	• Padrões das telas (cor, cabecalho e rodapé);		

Etapa	Descrição	Responsável	Registros Obrigatórios
Ambiente de desenvolvimento	<ul style="list-style-type: none"> • Hierarquia de navegação; • Dinâmica dos componentes de tela; • Alarmes; • Lógica de intertravamento e segurança; • Níveis de segurança da aplicação; • Relatórios. 	Equipe do Projeto	Não há
Protótipo	<p>Montagem do ambiente de desenvolvimento.</p> <p>Desenho de telas, navegação do sistema.</p>	Equipe do Projeto	Não há
Configuração do Sistema	<p>Desenvolvimento da interface gráfica:</p> <ul style="list-style-type: none"> • Kits de animação de válvulas, equipamentos, controladores PID, barras de nível; • Base de dados e configuração dos drivers de comunicação; 	Equipe do Projeto	Não há

Etapa	Descrição	Responsável	Registros Obrigatórios
	<ul style="list-style-type: none"> • Animação das telas; • Telas de alarmes 		
	<p>Configuração da lógica de intertravamento de processo e segurança:</p> <ul style="list-style-type: none"> • Kits de tratamento de sinais de válvulas, equipamentos, controladores PID, etc; • Tratamento de entradas analógicas; • Lógica de intertravamento de processo e segurança. 		
Teste em bancada	Testes integrados entre a interface gráfica e a lógica de processo segurança que roda no dispositivo de controle	Equipe de projeto	Relatório de testes
Treinamento	Capacitação da equipe de operação da planta às etapas de pré-operação e partida.	Equipe de projeto	Manual de operação;
			Questionário de avaliação do treinamento.

Etapa	Descrição	Responsável	Registros Obrigatórios
Implantação do Sistema e Operação Assistida	<p>Restauração/ reprodução do ambiente de desenvolvimento no ambiente de produção;</p> <p>Teste das malhas de controle;</p> <p>Teste da lógica de intertravamento;</p> <p>Operação assistida da planta</p>	Equipe do projeto	<p>Relatório dos testes de verificação da instalação do sistema;</p> <p>Relatório dos testes das malhas de controle;</p> <p>Relatório dos testes de intertravamento;</p> <p>Relatório de ocorrências da partida.</p>

5.3.3 Problemas do processo

O processo descrito no item 5.3.2 pode ser empregado em qualquer projeto de automação e controle. Por ser genérico, ele apresenta algumas deficiências para o projeto analisado, o qual possui as seguintes características:

- Projeto que envolve somente a construção de lógicas de intertravamento e controle regulatório no sistema de controle (CLP), ele não envolve a construção de telas (interfaces) em sistemas supervisórios;
- Não se trata de um projeto novo, de uma nova etapa do processo ou equipamento. É um projeto de melhoria, as lógicas desenvolvidas correspondem a modificações nas lógicas existentes, de modo que elas tiveram que ser inseridas no CLP de produção em paralelo às lógicas que estavam em funcionamento;
- Geralmente, nos projetos de automação e controle, as lógicas desenvolvidas se encaixam em 2 categorias principais:
 - Lógicas de intertravamento de segurança;
 - Lógicas de controle regulatório.

Este projeto engloba estas 2 categorias de lógica e também envolve a lógica de movimentação de equipamentos que é bem mais complexa do que as apresentadas anteriormente.

- Este projeto envolve o desenvolvimento de lógicas para diversas etapas do processo de produção industrial da planta, envolvendo diferentes linhas de produção e diferentes CLPs (com versões diferentes).
- Além do desenvolvimento de lógicas nos CLPs, também foi utilizado um sistema especialista para a otimização de algumas ações. Os sistemas especialistas são programas constituídos por uma série de regras, que analisam condições de entrada para solucionar um problema de um domínio conhecido.

Considerando as características deste projeto apresentadas anteriormente, podem ser destacados os seguintes problemas no processo existente:

- O processo está muito focado para projetos de novas unidades ou equipamentos, não estando preparado para projetos de melhoria;
- Este processo aborda somente o desenvolvimento de lógicas em CLPs ou SDCDs, não abordando sistemas especialistas;
- Os produtos a serem desenvolvidos nos projetos de automação e controle não possuem um padrão, que define quais os diagramas que devem ser desenvolvidos, qual a linguagem a ser adotada, entre outros;
- Segundo este processo, toda a implementação do sistema será realizada na etapa final do processo o que pode gerar vários problemas:
 - Atraso do projeto;
 - Um trabalho maior para consertar os erros;
 - Maior dificuldade para o desenvolvimento de lógicas interligadas;
 - Maior dificuldade de gerenciar os requisitos;
 - Maior risco.

5.4 Aplicação da proposta

Inicialmente, o projeto analisado neste estudo de caso seguia o processo de desenvolvimento de sistemas de automação e controle definido para a empresa apresentado no item 5.3, porém devido aos problemas mencionados no item 5.3.3, o processo de desenvolvimento deste projeto foi modificado, adotando o processo descrito no item 4, que propõe uma adaptação do processo unificado para projetos de automação e controle.

O projeto engloba quatro etapas do processo de fabricação da planta, os quais possuem necessidades especiais, de modo que para cada etapa do processo industrial foram definidas ações, que podiam ser interdependentes ou não. As ações correspondem a todo o processo desde o planejamento até a implantação de um

determinado conjunto de requisitos. Ao todo foram definidas 30 ações, conforme apresentado na Figura 10.

Para a etapa do processo A foram definidas 7 ações, das quais 6 ações eram interdependentes e seqüenciais. Já para a etapa do processo B, as suas 7 ações podem ser divididas em 2 grupos de ações seqüenciais, um grupo com 5 ações e outro com 2 ações, este último depende da ação 7 da etapa do processo A.

Os processos C e D possuem 8 ações cada e ambos estão divididos em 2 grupos de ações seqüenciais, onde o primeiro grupo engloba as ações de 1 a 6 e o segundo grupo, as ações 7 e 8. Existe uma dependência entre as ações 1, 2 e 7 dos dois processos, de modo que estas ações do processo D somente serão executadas após as ações do processo C.

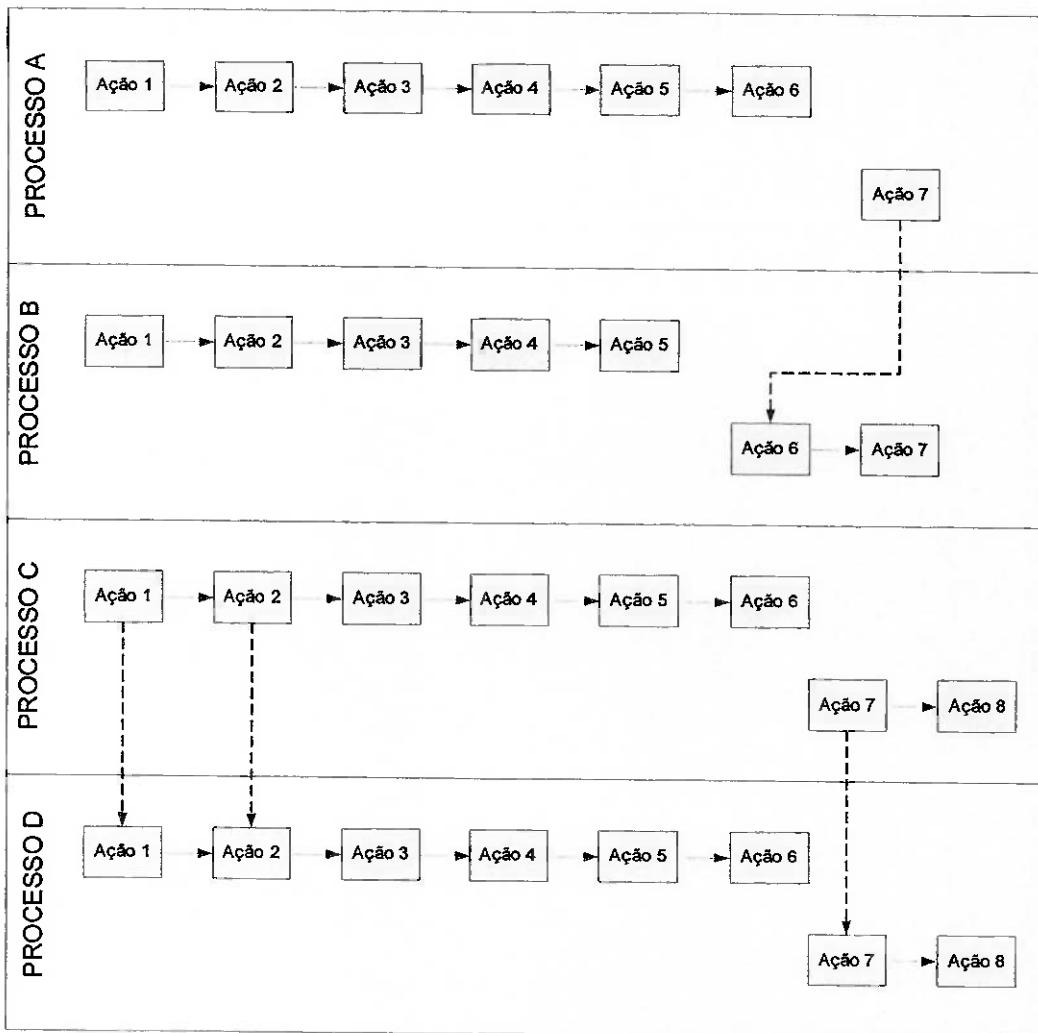


Figura 10 – Ações definidas no projeto e suas dependências.

Cada ação corresponde a uma iteração do processo. As ações são formadas por etapas, que podem ser associadas às disciplinas do RUP: requisitos, análise e projeto, implementação, teste e implantação (Tabela 6).

As demais disciplinas do RUP (modelagem do negócio, gerenciamento de configuração e mudanças, gerenciamento do projeto e ambiente) foram executadas ao longo de todo projeto e possuem as mesmas atribuições definidas pelo RUP.

Tabela 6 – Etapas das ações.

Etapa da ação	Descrição da etapa	Disciplina do RUP
Planejamento	<ul style="list-style-type: none"> • Especificação da lógica a ser implementada; • Levantamento dos riscos e ações para a sua mitigação; • Levantamento dos cenários de teste; 	Requisitos Análise e Projeto
Desenvolvimento	<ul style="list-style-type: none"> • Construção da lógica 	Implementação
Testes	<ul style="list-style-type: none"> • Testes integrados no ambiente de desenvolvimento, utilizando ferramentas de simulação (do próprio Software de programação do CLP, MatLab e Excel); 	Teste

Etapa da ação	Descrição da etapa	Disciplina do RUP
Homologação	<ul style="list-style-type: none"> • Homologação da lógica com o cliente no ambiente de desenvolvimento, através da simulação das variáveis do campo. A homologação foi desenvolvida em 2 níveis: <ul style="list-style-type: none"> ○ Automação: validação da lógica, forma de programação, nomenclatura das variáveis, utilização de padrões; ○ Operação: validação da funcionalidade da lógica através do seu acompanhamento nas telas do sistema supervisório; 	Teste
Modificações no CLP e no sistema supervisório (atividades realizadas pelo cliente)	<ul style="list-style-type: none"> • Transcrição das lógicas para o ambiente de produção (CLP de produção ou sistema especialista de produção); • Modificações nas telas do sistema supervisório. 	Implantação
Partida	<ul style="list-style-type: none"> • Ativação da lógica; • Configuração e sintonia de parâmetros. 	Implantação

Etapa da ação	Descrição da etapa	Disciplina do RUP
Operação Assistida	<ul style="list-style-type: none"> • Acompanhamento da lógica no ambiente de produção 	Implantação

5.4.1 Pontos da metodologia

Os 10 pontos da metodologia RUP propostos no item 4.2 foram aplicados neste estudo de caso. Fez-se uma análise da aplicação destes pontos, através do levantamento de aspectos positivos e negativos e de sugestões de melhorias:

- **Desenvolver uma visão:** É uma atividade obrigatória de todos os projetos, tendo como resultado a especificação funcional do sistema, porém os casos de uso não foram desenvolvidos. A construção dos casos de uso deveria ser obrigatória.
- **Gerenciar o plano:** Em todo projeto da Chemtech é desenvolvido o Plano do Projeto, para o qual existe um template. O plano de projeto é complementado pelo MS Project, apresentando o cronograma do projeto, alocação dos recursos, caminhos críticos, além do WBS. Os projetos são desenvolvidos segundo os conceitos do PMI.
- **Identificação e mitigação dos riscos:** Os riscos foram identificados, documentados e classificados em um portal. Não há uma metodologia formal para a identificação e mitigação dos riscos. Os riscos são identificados de acordo com a experiência do Gerente ou do Líder de projeto, que os classifica informalmente na hora de definir as iterações.

Deveria ser criado um procedimento para identificação dos riscos que serão classificados e documentados, pode ser gerado um check-list para auxílio na identificação dos riscos e um template do documento de classificação de riscos. A Chemtech possui uma matriz de análise de riscos que relaciona a criticidade com a probabilidade. Esta matriz é utilizada nos projetos de análise de risco e poderia ser incorporada ao processo proposto.

- **Gerar e controlar versões:** As versões do projeto foram controladas via Source Safe, porém muitas vezes elas não foram rotuladas. Tem-se a necessidade de definição de um procedimento para rotulação e controle formal das versões.
- **Examinar o caso de negócio:** A análise foi feita para determinar milestones de pagamentos, garantindo o fluxo de caixa.
- **Desenvolver a arquitetura dos componentes:** A arquitetura de Sistema foi apresentada na Especificação Técnica do Sistema.
- **Construir e testar o produto incrementalmente:** O produto foi continuamente testado durante o processo de desenvolvimento. Foram utilizadas ferramentas automáticas para a realização dos testes unitários, realizando simulações através do Excel e do MatLab. Em cada iteração foi gerado um executável (lógica), que geralmente foi apresentado ao cliente.
- **Verificar e avaliar os resultados:** Foi realizado um plano de testes no início do projeto. Gerou-se o documento de Resultado dos testes.
- **Gerenciar e controlar modificações:** Todos os documentos gerados durante o projeto são armazenados no Source Safe, sendo possível controlar as suas versões, porém não é feito um controle formal das modificações dos requisitos. Muitas vezes o requisito é diretamente modificado na especificação, sem referenciar como era antes, outras vezes, a mudança é incorporada ao projeto, mas não é devidamente documentada. O controle e o gerenciamento de requisitos são fundamentais. As pessoas devem se conscientizar de que toda modificação deve ser documentada.
- **Dar suporte ao usuário:** Foi desenvolvido o Manual de Instalação, Configuração e Operação do Sistema. Toda modificação foi apresentada à operação logo após a sua implementação e o documento foi sendo incrementado a cada partida de uma nova ação. Convém ressaltar que o controle da versão não é formal, assim não é apresentado os Release Notes.

5.4.2 Frameworks criados

A fim de tornar o desenvolvimento mais ágil, foram criados 3 frameworks, de acordo com a linguagem de programação ou o sistema no qual foi implementado (CLP, SDCD ou sistema especialista):

- Lógicas de CLP em Ladder;
- Lógicas de CLP em FBD;
- Lógicas no sistema especialista.

Nestes frameworks, foram definidos:

- Templates para lógicas:
 - Lógicas de controladores do tipo PID (proporcional, integral e derivativo);
 - Acionamento de equipamentos;
 - Intertravamentos.
- Padrões de nomenclatura.

5.4.3 Papéis

Considerando a equipe envolvida no projeto, foram definidos os papéis apresentados na Tabela 7, segundo proposto no item 4.4. Todos os papéis propostos no processo definido foram empregados neste projeto. O líder de projeto além de desempenhar o seu papel de líder, também teve a atribuição de arquiteto do sistema.

Tabela 7 – Mapeamento papéis.

Papel no Processo Proposto	Membro da equipe
Gerente de Projetos	Gerente de Projetos
Líder de Projeto	Líder de Projetos
Arquiteto do Sistema	Líder de Projetos
Projetista da Lógica	Consultor e engenheiro sênior
Projetista da Interface	Consultor e engenheiro sênior
Programador	1 Engenheiro pleno e 3 engenheiros júnior
Testador	1 Engenheiro pleno e 1 engenheiro júnior
Assistente administrativo	Secretária de projetos

5.4.4 Artefatos

Esta sessão compara os artefatos gerados neste estudo de caso e os propostos no item 4.5.

Tabela 8 – Artefatos propostos e realizados.

Artefato da Proposta	Construído no Estudo de caso	Comentário
Especificação funcional do sistema	Sim	É obrigatório a todos os projetos da empresa.
Plano de Projeto	Sim	Todos os projetos da Chemtech possuem o plano de projeto.

Artefato da Proposta	Construído no	Comentário
	Estudo de caso	
Casos de uso	Sim	No início do projeto, adotou-se o processo tradicional de projetos de automação e controle, que não prevê os casos de uso. Durante o projeto, alterou-se o seu processo de desenvolvimento e sentiu-se a necessidade dos casos de uso.
Protótipo	Não	Este é um projeto de melhorias, de modo que um protótipo não se fez necessário.
Diagrama de estado	Não	Por se tratar o primeiro projeto a adotar este processo, ainda não havia maturidade suficiente para a construção destes diagramas.
Diagrama de seqüência	Não	
Diagrama de atividade	Não	
Documento de arquitetura	Não	Como se trata de um projeto de melhoria, a arquitetura do sistema já estava definida. Porém, desenvolveu-se um documento de arquitetura para o ambiente de desenvolvimento e teste.
Checklist do teste unitário	Sim	Foram definidos cenários de teste, que foram executados

Artefato da Proposta	Construído no Estudo de caso	Comentário
Plano de testes e casos de teste		teste, que foram executados com o cliente para a validação do desenvolvimento.
Relatório de teste		
Manual de Operação; Manual de instalação e Configuração	Sim	Para cada ação desenvolvida, foi construído um manual de operação. O manual de instalação e configuração não foi escrito, pois é um projeto de melhoria e não foram introduzidas modificações aos manuais existentes.
Documento com o aceite do cliente	Sim	Após o período de operação assistida de cada ação, foi solicitado o seu aceite.

Conforme é possível observar pela Tabela 8, grande parte dos artefatos propostos pelo processo foi desenvolvida neste estudo de caso. Alguns deles não foram construídos, pois não se mostraram essenciais a este projeto, como é o caso do protótipo e do documento de arquitetura. Já os diagramas UML propostos (estado, atividade e seqüência) não foram utilizados, pois este processo foi adotado no meio do andamento do projeto e a equipe não estava familiarizada a ele, estes diagramas serão adotados nos projetos futuros e padrões serão definidos para os mesmos.

5.4.5 Comparação entre o processo proposto e os processos definidos para a empresa analisada

A Chemtech possui processos definidos para projetos de automação e controle (descrito no item 5.3) e de desenvolvimento de Software (apresentado no Anexo 1).

Uma grande diferença entre o processo proposto e o processo de automação e controle da empresa analisada é que o primeiro é iterativo e incremental, enquanto o segundo é seqüencial, semelhante ao processo em cascata. Já o processo de desenvolvimento de Software da Chemtech e o processo proposto se baseiam no RUP.

Ao se compararem os artefatos gerados em cada um dos processos, nota-se uma grande semelhança entre eles, conforme apresentado na Tabela 9. A maior diferença reside nos diagramas UML propostos (classe, estado, atividade e seqüência) que são inexistentes no processo de automação e controle existente, porém são obrigatórios no processo de desenvolvimento de software, com exceção do diagrama de atividade.

O processo de automação e controle possui uma deficiência que é a inexistência de padrões para a modelagem do sistema, que pode ser suprida com a adoção de diagramas UML, conforme o processo proposto adota.

Os papéis e artefatos definidos nos processos são semelhantes. No processo proposto há a inclusão do papel do projetista da lógica, atividade que é particular dos projetos de automação e controle.

Tabela 9 – Comparação entre Artefatos dos processos existentes e proposto.

Artefato	Processo		
	Proposto	Software da Chemtech	Autom. e controle da Chemtech
Especificação funcional do sistema	X	X	X
Plano de Projeto	X	X	X

Artefato	Processo		
	Proposto	Software da Chemtech	Autom. e controle da Chemtech
Casos de uso	X	X	
Protótipo	X	X	X
Diagrama de classe		X	
Diagrama de estado	X	X	
Diagrama de seqüência	X	X	
Diagrama de atividade	X		
Documento de arquitetura	X	X	X
Teste de comunicação			X
Checklist do teste unitário	X	X	
Plano de testes e casos de teste	X	X	X
Relatório de teste	X	X	X
Manual de Operação	X	X	X
Manual de instalação e Configuração	X	X	X
Documento com o aceite do cliente	X	X	X

5.5 Considerações finais

O projeto analisado neste estudo de caso é um projeto de controle que possui como característica ser um projeto de melhoria, onde novas lógicas de controle tiveram que ser incorporadas à lógica existente.

Todo o seu processo de implantação teve que ser executado a quente, ou seja, sem a interrupção da lógica existente para a ativação da nova lógica, sendo um projeto extremamente crítico, pois uma falha na lógica poderia levar à parada da planta ou a algum acidente.

No início do projeto, adotou-se o processo de desenvolvimento de projetos de automação e controle definido nesta empresa, o qual se mostrou ineficiente e inadequado, uma vez que possuía um risco muito grande associado a sua implementação, já que considerava o desenvolvimento de todas as lógicas de controle, para então realizar os seus testes e implantação de uma só vez.

Uma vez detectados os problemas desta metodologia, adotou-se o processo proposto no item 4, que tem se mostrado bastante efetivo até a presente data.

Todas as lógicas foram divididas em ações, que correspondem a iterações. Ao final de cada iteração, as lógicas especificadas nas ações estavam devidamente implantadas e validadas em produção.

A adoção de um processo iterativo e incremental diminuiu os riscos do projeto e trouxe maior segurança à equipe de projeto e ao cliente. Também permitiu um gerenciamento mais efetivo dos requisitos.

O processo proposto adota grande parte dos artefatos definidos pelo processo de desenvolvimento de Software existente na empresa, aliado à metodologia RUP, que este processo já adota e é de conhecimento de todos os funcionários da empresa.

As peculiaridades dos projetos de automação e controle definidas no processo de desenvolvimento deste tipo de projetos também foram incorporadas ao processo proposto.

6 Conclusão

Este trabalho apresenta uma proposta de aplicação do RUP em projetos de automação e controle.

A metodologia de desenvolvimento de sistemas de controle enfoca mais a lógica a ser desenvolvida, possuindo ferramentas e diagramas que auxiliam a sua definição, porém ela desconsidera o processo de desenvolvimento como um todo, ou seja, tem uma grande deficiência no levantamento de requisitos, bem como na definição de etapas do processo.

Os projetos de automação e controle possuem um risco muito alto, pois, geralmente, eles são integrados a quente a uma aplicação que já está rodando, ou seja, o sistema atual não é parado para a inserção das novas lógicas, onde uma falha na lógica a ser implementada pode causar a parada de uma unidade ou até mesmo provocar acidentes, uma vez que as lógicas envolvem o controle de variáveis de um processo.

O RUP traz ferramentas para um desenvolvimento incremental e iterativo, que permite que pequenas porções de lógicas sejam implantadas no sistema de produção a cada iteração, diminuindo os riscos e tornando-os mais gerenciáveis.

A dinâmica do processo é uma variável difícil de ser gerenciada em um projeto de controle, de modo que a adoção de um processo incremental traz uma grande vantagem aos projetos de controle, uma vez que ele permite a análise dos resultados obtidos em uma iteração, antes da definição das próximas ações a serem tomadas nas iterações subsequentes.

Convém também ressaltar que o RUP emprega diagramas UML que são bem definidos e estruturados e podem auxiliar a preencher as lacunas existentes nos processos de desenvolvimento de projetos de automação e controle.

A adoção do processo proposto neste trabalho no projeto apresentado no estudo de caso solucionou os problemas que o processo existente apresentava e contribuiu para que o projeto se desenvolvesse conforme o planejado, seguindo os prazos, com riscos mais controlados, com maior flexibilidade e dando maior segurança à equipe do

projeto e ao cliente. Este projeto adotou parcialmente o processo proposto durante o seu andamento. As suas etapas foram empregadas completamente, porém, dentre os seus artefatos, não foram elaborados os diagramas propostos e ainda não foram definidos padrões para os mesmos.

6.1 Trabalhos futuros

A partir dos resultados apresentados nesta monografia, têm-se as seguintes frentes de trabalho:

1. O estudo de caso apresentado nesta monografia foi um trabalho preliminar. Frente ao seu resultado positivo, propõe-se a definição dos documentos padrões envolvidos nos projetos de automação e controle;
2. Apresentação dos resultados obtidos com este estudo de caso para a Chemtech, com uma proposta para a alteração do processo definido para os projetos de automação e controle. Uma vez aceito, será feita a análise para a recertificação destes processos com a ISO;
3. Conforme apresentado no início desta monografia, a Chemtech é uma empresa que realiza projetos de natureza bastante distinta. Atualmente, somente os projetos de desenvolvimento de software adotam o processo unificado, mais especificamente, o RUP. Uma proposta para os trabalhos futuros é a análise da adoção do processo unificado para outras áreas, como:
 - a. Sistema de informação;
 - b. Planejamento e programação de produção;
 - c. Sistema de informações laboratoriais (LIMS);
 - d. Visualização de dados.

Referência Bibliográfica

- BECK, K. **Extreme programming Explained**, Addison-Wesley, 2000.
- GONG, Y. **CORBA Application in Real-Time Distributed Embedded Systems, Survey Report**, ECE 8990 Real-Time Systems Design, Spring 2003.
- HECK, B. S.; WILLS, L. M.; VACHTSEVANOS, G. J. **Software Technology for Implementing Reusable, Distributed Control Systems**, IEEE Control Systems Magazine, Feb. 2003.
- KRISHNA, A. S.; SCHMIDT, D. C.; KLEFSTAD, R.; CORSARO, A. **Real-time CORBA Middleware, Middleware for Communications**, John Wiley & Sons, 2001.
- KRUCHTEN, P., KROLL, P. **The Rational Unified Process Made Easy**, Addison-Wesley, 2003.
- KRUCHTEN, P. **Agility with the RUP**, The Rational Edge, 2002.
- LEE, B. K.; SHIN, Y. H.; **An Upper Intelligent Control System Using the Object Oriented Method**, SICE-ICASE International Joint Conference, Oct. 2006.
- PANJAITAN, S.; FREY, G.; **Combination of UML Modeling and the IEC 61499 Function Block Concept for the Development of Distributed Automation Systems**, IEEE Conference on Emerging Technologies and Factory Automation, 2006.
- PROBASCO, L. **The Ten Essencials of RUP**, The Rational Edge, 2001.
- RATIONAL SOFTWARE CORPORATION **Rational Unified Process**, Version 2001.03.00. CD-ROM, Rational Software, Cupertino, California, 2001.

REIS, J. O.; BELCHIOR, A. D. **Myrup: uma Adaptação do RUP para Projetos de Pequeno e Médio Porte**, XXX Conferencia Latinoamericana de Informática CLEI, 2004.

SMTTH, J. **A Comparation of RUP and XP**, Rational White Paper, 2001.

THRAMBOULIDIS, K. **Using UML for the development of Distributed Industrial Process Measurement and Control Systems**, IEEE Conference on Control Applications (CCA), Sep. 2001.

THRAMBOULIDIS, K. **An Architecture to Extend the IEC61499 Model for Distributed Control Applications**, 7th International Conference on Automation Technology, (Automation 2003), 2003.

THRAMBOULIDIS, K. **Using UML in Control and Automation: A Model Driven Approach**, IEEE International Conference on Industrial Informatics, Volume 2, 2004.

TRANORIS, C.; THRAMBOULIDIS, K. **Integrating UML and the Function Block Concept for the Development of Distributed Control Applications**, IEEE Conference on Emerging Technologies and Factory Automation, Volume 2, Issue, 16-19 Sep. 2003.

WEHRMEISTER, M. A.; PEREIRA, C. E.; BECKER, L. B. **Object-Oriented Methodology to the Development of Embedded Real-Time Systems**, 3rd IEEE International Conference on Industrial Informatics, 2005.

Anexo 1: Processo atual de Projetos de Desenvolvimento de Software

Os projetos de automação e controle da Chemtech possuem o fluxo e processos de desenvolvimento apresentados nos itens a seguir.

Fluxo de desenvolvimento

A Figura 11, a Figura 12 e a Figura 13 apresentam o fluxo de desenvolvimento do projeto através das responsabilidades atribuídas ao gerente do projeto, ao líder do projeto e a equipe de desenvolvimento.

Conforme descrito no item 5.3.1, a fase inicial dos processos da Chemtech é comum aos projetos, assim para os projetos de desenvolvimento de software, inicialmente é realizada uma reunião com o Gerente Comercial e com o Líder do Projeto, na qual é apresentado o escopo do projeto, o prazo, custos, lucro estimado, restrições, entre outros. Após esta reunião é elaborado o plano do projeto, usando como ferramenta o MS Project, além do WBS. Após esta etapa, utilizando uma ferramenta da Intranet da Empresa, é definido o início do projeto.

O Líder do projeto escreve a Especificação Funcional e os Casos de Uso principais do Sistema. Após esta etapa, é realizada uma reunião com o cliente para a validação da Especificação Funcional, dos Casos de Uso e do Plano do Projeto. Em seguida, é realizada uma reunião com a equipe do projeto, na qual é apresentado o sistema a ser desenvolvido, o plano do projeto e as atribuições de cada membro da equipe.

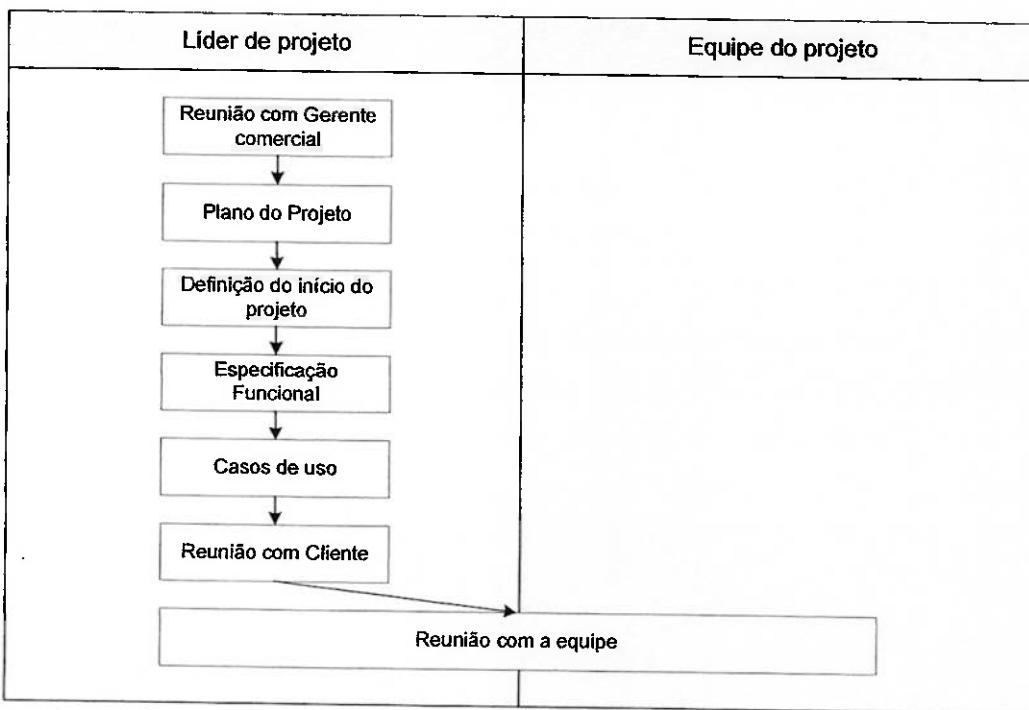


Figura 11 – Fluxo de desenvolvimento do Projeto – Etapa inicial.

Após a reunião com a equipe, o líder de projeto constrói o diagrama de classes, escreve a especificação técnica e o plano e a especificação de testes. Já a equipe de projeto prepara o ambiente de desenvolvimento e inicia o desenvolvimento do sistema, sempre acompanhado por testes unitários.

No final desta fase, são gerados os manuais de instalação, operação e configuração, além de serem realizados testes integrados no ambiente de desenvolvimento. Após os testes no ambiente de desenvolvimento, realizam-se testes no ambiente de testes do cliente. Todos os testes são realizados de acordo com o plano de testes.

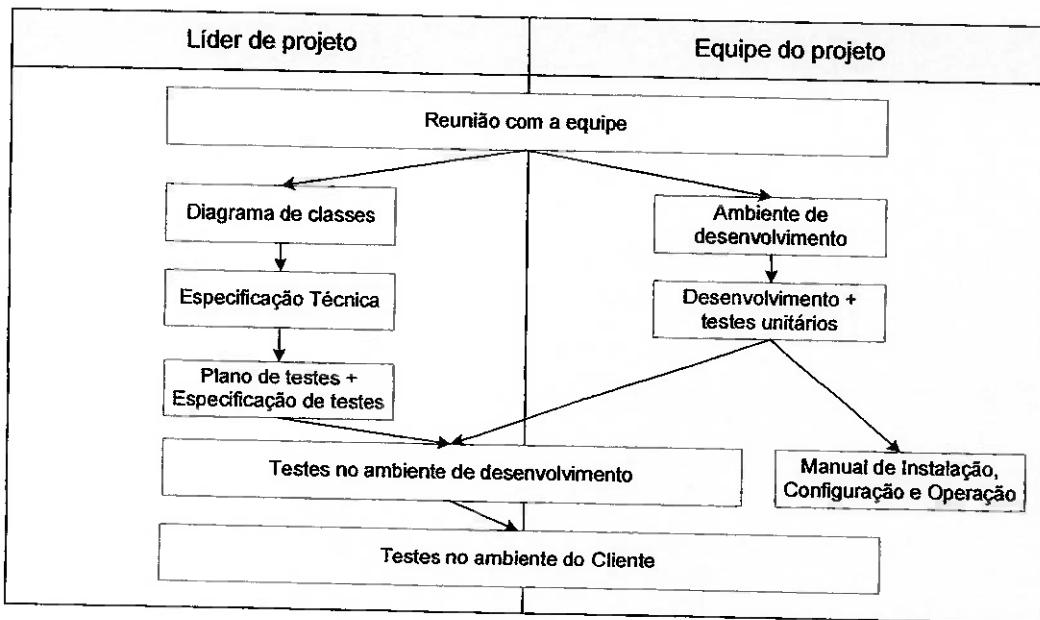


Figura 12 – Fluxo de desenvolvimento do Projeto – Etapa intermediária.

Uma vez validado com o cliente, são gerados os documentos com os resultados dos testes e o sistema é implantado no ambiente de produção e inicia um período de operação assistida. O projeto então recebe o aceite do cliente. Uma avaliação do projeto é completada pelo cliente e o projeto é fechado na Intranet da Chemtech.

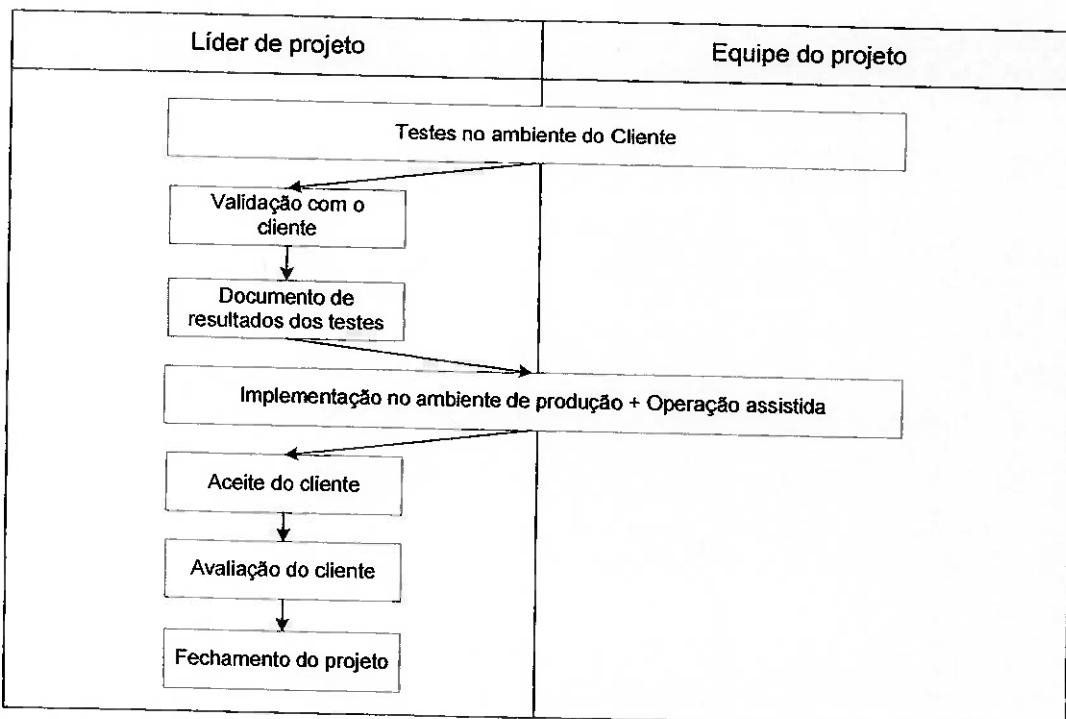


Figura 13 – Fluxo de desenvolvimento do Projeto – Etapa final.

Processo de desenvolvimento

O processo de desenvolvimento de Software utilizado pela Chemtech é o RUP.

A Tabela 10 apresenta as atividades a serem desenvolvidas no projeto, os artefatos de entrada, os documentos produzidos, os responsáveis pela atividade e se a atividade é obrigatória ou não.

Tabela 10 – Tabela de responsabilidades, atividades e artefato.

Atividade	Entrada	Documento produzido	Responsável	Obrigatório
Definir visão do projeto	Levantamento de requisitos	Especificação funcional do sistema	<ul style="list-style-type: none"> ➤ Gerente do projeto ➤ Líder de projeto ➤ Responsável do cliente 	Sim
Elaborar Plano do projeto	Especificação Funcional do Sistema	Plano de Projeto	<ul style="list-style-type: none"> ➤ Gerente do projeto ➤ Líder de projeto 	Sim
Detalhar Requisitos do Sistema	Especificação funcional do sistema e/ou Proposta Técnica	Casos de uso Descrição de casos de uso Protótipo de interface Diagrama de classe	<ul style="list-style-type: none"> ➤ Líder de projeto ➤ Equipe de projeto 	Não Não Não Sim

Atividade	Entrada	Documento produzido	Responsável	Obrigatório
		Diagrama de estado		Não
		Diagrama de seqüência		Não
Projetar arquitetura do sistema	Especificação Funcional do Sistema, casos de uso, diagramas de classe, diagramas de estado, diagramas de seqüência, protótipo de interface, modelo de dados, prova de conceitos	Documento de arquitetura	➢ Líder de projeto ➢ Equipe de projeto	Não
Codificar aplicação	a Especificação funcional do sistema, documento de arquitetura	Código fonte	➢ Equipe do projeto	Sim
Realizar teste unitário	Código fonte	Checklist do teste unitário	➢ Equipe do projeto	Não
Realizar testes de integração	Especificação funcional do sistema,	Plano de testes e casos de teste	➢ Líder do projeto	Não

Atividade	Entrada	Documento produzido	Responsável	Obrigatório
	documento de arquitetura, código fonte	Relatório de teste		Sim
Gerenciar configurações	Especificação funcional do sistema, solicitação de alteração, verificar quais documentos necessitarão ser alterados	Atualizar plano de projeto Código Fonte (utilizar ferramenta de controle de versões, ex Source Safe)	➤ Gerente de projeto ➤ Equipe do projeto	Sim
		Atualizar documento de arquitetura	➤ Líder do projeto	Não
Entrega do Sistema	Especificação funcional do sistema	Manual de Operação; Manual de instalação e Configuração	➤ Equipe do projeto	Sim
Treinamento	Manual de Operação; Manual de instalação e Configuração	Avaliação por parte do cliente sobre o treinamento	➤ Equipe do projeto	Sim

Atividade	Entrada	Documento produzido	Responsável	Obrigatório
Aceite do sistema		Documento com o aceite do cliente	➢ Gerente do projeto	Sim

SILVIO SHINJI YOSHINO

**ANÁLISE DO WINDOWS COMMUNICATION FOUNDATION ATRAVÉS
DO CONCEITO DE COMPOSIÇÃO DO SEI PARA O
DESENVOLVIMENTO BASEADO EM COMPONENTES**

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Pós-Graduação em Tecnologia de Software.

Área de Concentração:
Engenharia de Software

Orientador:
Profº Doutor
Paulo Sérgio Muniz Silva

São Paulo

2007