

**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO**

**Relatório de Estágio
Gestão de Serviços de Valor Agregado (GVAS)**

Tiago de Lima Wanderley
Estagiário

Joseluze Farias Cunha
Orientadora Acadêmica

Francisco de Barros Neto
Supervisor Técnico

Campina Grande

Julho de 2009

TIAGO DE LIMA WANDERLEY

Gestão de Serviços de Valor Agregado (GVAS)

Orientadora: Joseluze Farias Cunhas

Relatório de estágio apresentado como parte das exigências do curso de Ciência da Computação, da Universidade Federal de Campina Grande, para cumprimento da disciplina de estágio integrado.

Campina Grande

Julho 2009



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

Gestão de Serviços de Valor Agregado (GVAS)

Aprovado Em _____

BANCA EXAMINADORA

Professora Mestre Joseluze Farias Cunha
ORIENTADORA ACADÊMICA

Professor Mestre Bruno Correia da Nóbrega Queiroz
MEMBRO DA BANCA

Professora DRa. Joseana Macêdo Fachine
MEMBRO DA BANCA

Campina Grande

Julho de 2009

À
minha querida esposa,
meu adorável filho,
meu pai,
minha mãe
e todos os familiares e amigos que
de alguma forma contribuíram
para a elaboração deste relatório.

Agradecimentos

Agradeço imensamente a Accenture por ter me dado esta oportunidade, especialmente a Júlio e a Jaílson. Também, gostaria de agradecer a minha orientadora Joseluce pela paciência na orientação e incentivo que tornaram possível a conclusão deste relatório. Gostaria de agradecer a professora Joseana Fachine por ter me fornecido o embasamento da função do estagiário, de como elaborar o relatório e de ter ministrado a disciplina no período que estive cursando a mesma. Também agradeço ao professor Bruno por ter aceitado o convite para fazer parte da banca e ao meu supervisor técnico Francisco por ter me auxiliado durante o desenvolvimento do projeto e elaboração do relatório.

Resumo

Cada vez mais, o celular deixa de ser uma simples ferramenta de comunicação por voz, sendo utilizado para o envio e recebimento de emails, download de vídeos, música, imagens, recebimento de notícias, etc.

Como grande parte destes novos serviços é provida por plataformas de rede gerenciadas por outras empresas, a Empresa PHONEMAX deve repassar a estas empresas parte do valor arrecadado com a utilização destes serviços por seus usuários.

O sistema GVAS é responsável pelo gerenciamento financeiro e de repasse aos provedores pela prestação de serviços, mediante as regras definidas pelo contrato estabelecido entre a Operadora de telefonia e o fornecedor.

Neste sistema deverão ser registrados todos os serviços utilizados por clientes da Empresa PHONEMAX que tenham sido providos por plataformas gerenciadas por empresas terceiras, acompanhando também o ciclo de vida destes serviços (faturamento, arrecadação, contestação, rejeição etc.).

O GVAS é composto por um módulo Online, através do qual podem ser cadastrados os fornecedores, serviços e contratos, configurados os tipos de serviços, plataformas e usuários e gerados relatórios gerenciais, e um módulo batch, responsável por registrar na base de dados a utilização destes serviços por clientes da PHONEMAX.

A empresa de telefonia identificou a necessidade de inclusão de novas funcionalidades no GVAS, gerando impacto nos módulos Batch, Relatórios e OnLine. Tais funcionalidades geraram a necessidade de desenvolver regras de repasse e ressarcimento, configurações de contratos e serviços, aplicáveis a “serviços diferenciados”, independentemente de tráfego de dados.

Sumário

1	Introdução	1
2	Ambiente do Estágio	3
2.1	A instituição	3
2.2	Infra-Estrutura	3
2.3	Supervisão	4
3	Fundamentação Teórica	6
3.1	Novas Tecnologias Utilizadas no Estágio	7
3.1.1	Servidor Weblogic 8.1	8
3.1.2	Struts 1.2	8
3.1.3	Hibernate	8
3.1.4	Spring	9
3.1.5	Tiles	9
4	Atividades do Estágio	10
4.1	Metodologia de Desenvolvimento	10
4.2	Dificuldades Encontradas Pela Equipe	14
4.3	Atividades Planejadas	14

4.4	Atividades Realizadas	15
4.4.1	Novas Tecnologias	15
4.4.2	Coding Unit Test (CUT)	16
4.4.3	Peer Review	17
4.4.4	Assembly Test Preparation (ATP)	19
4.4.5	Assembly Test Execution (ATE)	19
5	Resultados esperados x Resultados obtidos	23
5.1	Módulos criados	23
5.2	Resultados Gerais Obtidos	24
5.3	Resultados esperados do estagiário	24
5.4	Resultados obtidos do estagiário	24
5.5	Desempenho do Estagiário	25
6	Considerações Finais	26
A	Plano De Estágio	29
B	Hibernate	47
C	Processo de Desenvolvimento	66
C.1	ADM	66
C.2	CMMi	66
C.3	V-Model	67
C.4	Metodologia Desenvolvimento CoE	68
C.5	Benefícios	69

C.6 QPI	69
C.7 Business Continuity Approach	70
C.8 Communication Plan	70
C.9 Configuration Management Plan	70
C.10 Knowledge Management Plan	70
C.11 Quality Management Plan	70
C.12 Risk Management Plan	71
C.13 Unit Operating Model	71
C.14 Unit Measurement Plan	71

Lista de Tabelas

5.1	Atividades de Cadastro realizadas e seus respectivos status	23
5.2	Telas criadas e seus respectivos status	23
5.3	Módulos criados e seus respectivos status	24
5.4	Resultados esperados do estagiário	24

Lista de Figuras

1.1	Arquitetura do GVAS	2
4.1	Framework V-Model	11
4.2	Diagrama de Atividades de Desenvolvimento - Parte 1	12
4.3	Diagrama de Atividades de Desenvolvimento - Parte 2	13
4.4	Diagrama de Atividades de Desenvolvimento - Parte 3	13
4.5	Diagrama de Atividades do Peer Review Parte 1	18
4.6	Diagrama de Atividades do Peer Review Parte 2	18
4.7	Diagrama de Atividades do Assembly Test Execution	20
4.8	Project Parte 1	21
4.9	Project Parte 2	22
C.1	Níveis de Maturidade	67
C.2	Framework V-Model	68

Capítulo 1

Introdução

Este relatório descreve e detalha todas as atividades desenvolvidas durante o estágio do aluno Tiago de Lima Wanderley do curso de Ciência da Computação da Universidade Federal de Campina Grande na Empresa de Outsourcing Accenture durante o período de 24/Março/09 a 05/Junho/09. Tal documento é parte fundamental da avaliação da disciplina Estágio Integrado [Fechine 2009] ministrada pela professora Joseana Macêdo Fechine [Fechine 2009], durante o semestre letivo 2009.1.

Diferentemente de alguns estágios, um período de estágio na Accenture pode proporcionar ao estagiário uma excelente chance de pôr em prática todos os conhecimentos adquiridos na universidade da forma que o mercado de trabalho exige. Por ser uma empresa firmada no mercado e, portanto, com metas e níveis de excelência a manter, o estagiário é mais exigido para cumprir seus deveres. O estagiário permanece, desde o início, em contato com a prática da atividade para a qual está se formando, o que lhe agrega uma experiência importante para o futuro no mercado de trabalho.

Esse período de estágio na empresa foi desenvolvido juntamente com mais três pessoas. O objetivo é modificar um sistema para atender às novas exigências impostas pela empresa de telefonia.

O estágio trata do desenvolvimento do GVAS. Módulo este, responsável, pelo gerenciamento financeiro e de repasse aos provedores pela prestação de serviços, mediante as regras definidas pelo contrato estabelecido entre a PHONEMAX e o fornecedor.

A arquitetura do sistema pode ser observada na Figura 1.1.

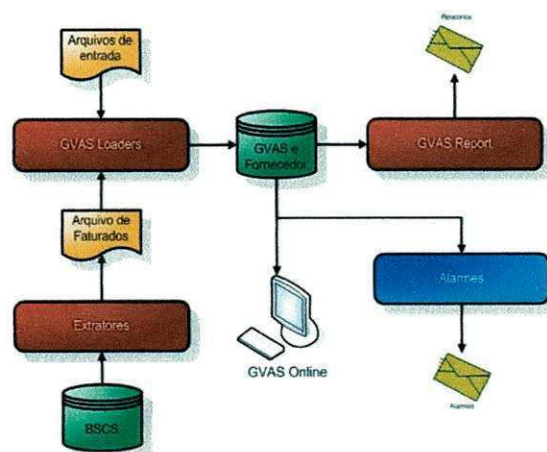


Figura 1.1: Arquitetura do GVAS.

O sistema GVAS antes de ser alterado, funcionava de maneira parecida com o sistema atual, só que ao contrário do que foi implementado, este, apenas contabilizava serviços da própria empresa como por exemplo, papéis de parede, ringtones e etc.

Por motivos de privacidade, o nome da empresa relacionada com o projeto será preservado e durante este relatório a mesma será chamada de PHONEMAX.

O projeto surgiu quando a PHONEMAX identificou a necessidade de inclusão de novas funcionalidades no GVAS, gerando impacto nos módulos Batch, Relatórios e OnLine. Tais funcionalidades geraram a necessidade de desenvolver regras de repasse e ressarcimento, configurações de contratos e serviços, aplicáveis a “serviços diferenciados”¹, independentemente de tráfego de dados.

Para atender a solicitação da PHONEMAX, o sistema GVAS será alterado para contemplar o repasse para os novos serviços diferenciados. Resumidamente, os seguintes processos serão criados/alterados:

- GVAS Online
- GVAS Batch
- GVAS Relatórios

¹Serviços de Terceiros na Fatura da PHONEMAX

Capítulo 2

Ambiente do Estágio

2.1 A instituição

A Accenture é uma consultoria global de gestão de serviços de tecnologia e outsourcing.

Comprometida com a inovação, a Accenture colabora com seus clientes ajudando-os a tornarem-se empresas ou governos de alto nível de performance.

Com mais de 186 mil empregados espalhados por 220 escritórios em 52 países, a Accenture teve receitas líquidas de US\$ 23.39 bilhões no ano fiscal encerrado em 31 de Agosto de 2008. O ano de 2009 iniciou com mais de 6000 funcionários no Brasil.

2.2 Infraestrutura

A execução das atividades do estágio ocorreu no Accenture, situada na Rua Maria Vieira César, número 193, no bairro Jardim Tavares, na cidade de Campina Grande, Paraíba.

O projeto foi desenvolvido em quatro máquinas de uso pessoal para cada integrante da equipe. Cada máquina é composta pelos seguintes (mais relevantes) componentes de hardware:

- Processador Intel® Pentium® 4 CPU 2.26 GHz;

- Dois módulos de memória RAM DDR SDRAM-400 de 1GB de capacidade;
- Dispositivo de armazenamento secundário Samsung® modelo HD160JJ/P de 160GB de capacidade;
- Adaptador de rede SiS 900 Fast Ethernet Adapter
- Monitor LCD AOC® de 17" modelo 712Sa.

Todas as máquinas estão interligadas em uma estrutura de rede cabeada de 100Mbps e têm acesso externo à Internet por meio de uma conexão de 1Mbps dedicado provida pelo serviço da Embratel. Em relação aos componentes de software instalados nas máquinas, abaixo estão listados os mais relevantes deles:

- Microsoft® Windows® XP Professional;
- Eclipse® Plataform 3.3.2;
- MySQL® Server 5.0;
- Java 1.4
- JDBC
- Servidor Weblogic 8.1
- Struts 1.2
- Hibernate
- Spring
- Tiles

2.3 Supervisão

Supervisora Acadêmica

Nome: Joseluze de Farias Cunha

Endereço: Departamento de Sistemas e Computação

Universidade Federal de Campina Grande

Avenida Aprígio Veloso, 882, CEP:58.109.970

Bodocongó, Campina Grande, PB, Brasil.

Email: joseluce@dsc.ufcg.edu.br

Supervisor Técnico

Nome: Francisco Manoel de Barros Neto

Endereço: Rua Sinhazinha Celino, 65, CEP: 58105.613

Distrito Industrial, Campina Grande, PB Brasil

Email: kiko.ufcg@gmail.com

Capítulo 3

Fundamentação Teórica

O GVAS é composto por três módulos: GVAS batch, GVAS On-line e GVAS Relatórios. Dentre estes estão as seguintes funcionalidades:

- GVAS Batch
 - Loader Pré-pago de serviços de VAS debitados na OPSC;
 - Extrator Pós-pago de serviços de VAS faturados com sucesso no BSCS;
 - Loader Pós-pago de serviços de VAS faturados com sucesso no BSCS;
 - Loader Pós-pago de serviços de VAS inadimplentes e arrecadados com sucesso no BSCS e RMCA;
 - Loader Pós-pago de serviços de VAS rejeitados ou considerados fraudulentos no BSCS;
 - Loader SDP de serviços de VAS entregues para clientes Pré-pago e Pós-pago;
 - Loader TDMA de serviços VAS entregues para clientes TDMA Pré-pago e Pós-pago;

- GVAS Online
 - Autenticação de usuário;
 - Controle de perfil;
 - Cadastro de fornecedores;
 - Cadastro de serviços;
 - Cadastro de contratos;

- Configuração de tipos de serviço;
- Configuração de plataformas;
- Configuração de usuários;
- GVAS Relatórios
 - Relatório de repasse;
 - Relatório de histórico de repasse;
 - Relatório de status;
 - Relatório de rejeição;
 - Relatório de lote;
 - Relatório de serviços sem contrato;
 - Relatório de serviços não cadastrados;
 - Relatório de relacionamento entre Contratos x Serviços x Fornecedores;
 - Relatório de repasse TDMA;

Para implementação das funcionalidades citadas, foi preciso fazer um estudo das tecnologias que eram parcialmente e/ou integralmente desconhecida pelos integrantes da equipe. O estagiário desconhecia integralmente as seguintes tecnologias:

- Spring;
- Tiles.

E parcialmente, as seguintes:

- Hibernate;
- Struts 1.2.

3.1 Novas Tecnologias Utilizadas no Estágio

Os próximos sub-tópicos detalham as mais relevantes tecnologias e ferramentas utilizadas nas atividades do estágio.

3.1.1 Servidor Weblogic 8.1

A Weblogic, Inc., foi fundada por Paul Ambrose, Bob Pasker, Laurie Pitman, e Carl Resnikoff, em Setembro de 1995. Antes disso, Ambrose e Resnikoff trabalharam no desenvolvimento (pre-JDBC) dos drivers Oracle, Sybase, e Microsoft SQL Server para Java com o nome de dbKona, assim como um servidor de “três camadas” (three tier) para permitir que applets pudessem conectar-se a esses bancos de dados. Este foi o servidor WebLogic 1.48, e era chamado T3Server (uma referência ao termo 3-Tier Server). Enquanto isso, Pitman and Pasker estavam trabalhando em ferramentas de gerenciamento de redes no Java. Pasker escreveu uma pilha SNMP em Java e métodos Win32 nativos para ICMP ping, ao passo que Pitman trabalhava em applets que mostrassem informações sobre gerenciamento (management data). Uma das funcionalidades ocultas da versão 1.48 do servidor era a habilidade de estender-se modificando um dispatcher e adicionando um handler para diferentes tipos de mensagens. Pasker falou com Ambrose sobre enviar um código fonte para o servidor, e Pasker estendeu este de forma que as applets pudessem fazer requisições de rede SNMP e PING, e mostrassem os resultados. A partir deste ponto, os fundadores trabalharam juntos para construir o que veio eventualmente a ser chamado de Application Server [WIKIPEDIA].

3.1.2 Struts 1.2

Struts é um framework de desenvolvimento da camada controladora, numa estrutura seguindo o padrão Model 2 (uma variante do MVC oficializada pela Sun), de aplicações web (principalmente) construído em Java para ser utilizado em um container web em um servidor J2EE. Este framework foi originalmente desenvolvido por Craig McClanahan e doado para a Apache Software Foundation, onde continua sendo desenvolvido segundo o padrão desta fundação.

3.1.3 Hibernate

Hibernate [HIBERNATE] é uma biblioteca open source de mapeamento objeto-relacional muito utilizada em conjunto com a linguagem Java e que provê um framework de manipulação de dados do sistema em uma base de dados relacional. Em outras palavras, Hibernate atua como mediador no mapeamento de classes Java em tabelas do banco de dados.

No desenvolvimento de um sistema de médio ou grande porte, o uso de Hiber-

nate é de fundamental importância, pois é ele que fica encarregado de gerar as chamadas SQL ao banco de dados, liberando o desenvolvedor da tarefa de manipular a conversão de objetos e as consultas e resultados de retorno manualmente. Essa metodologia torna a aplicação portátil para qualquer banco de dados SQL.

3.1.4 Spring

O Spring é um framework open source criado por Rod Johnson e descrito em seu livro “Expert One-on-One: JEE Design e Development”. Trata-se de um framework não intrusivo, baseado nos padrões de projeto inversão de controle (IoC) e injeção de dependência, esta é a maneira de remover as dependências entre os objetos “injetando” as referências para os colaboradores sobre demanda (Dependency Injection). O framework de IoC determina como os objetos serão injetados, normalmente através de um arquivo de configuração XML.

No Spring o container se encarrega de “instanciar” classes de uma aplicação Java e definir as dependências entre elas através de um arquivo de configuração em formato XML. Dessa forma o Spring permite o baixo acoplamento entre classes de uma aplicação orientada a objetos. O Spring possui uma arquitetura baseada em interfaces e POJOs (Plain Old Java Objects), oferecendo aos POJOs características como mecanismos de segurança e controle de transações. Também facilita testes unitários e surge como uma alternativa à complexidade existente no uso de EJBs. Esse framework oferece diversos módulos que podem ser utilizados de acordo com as necessidades do projeto, como módulos voltados para desenvolvimento Web, persistência, acesso remoto e programação orientada a aspectos[WIKIPEDIA].

3.1.5 Tiles

A framework Tiles [Tiles], que pode ser utilizada em conjunto com Struts, é uma implementação de composite view que faz uso de custom tags e provê toda a flexibilidade de reaproveitamento e gerenciamento de layout que o padrão propicia. Com Tiles, além do reuso das “partes” é possível também definir um padrão para a disposição destas nas páginas, ou seja, definir o layout que as organizam. Basicamente, define-se uma template com o layout padrão que um conjunto de páginas terá [Braz].

Capítulo 4

Atividades do Estágio

Nesta seção são apresentadas as atividades planejadas e realizadas no transcorrer do estágio. É importante destacar que as atividades desenvolvidas pelo estagiário, foram realizadas em módulos distintos.

Todas as atividades planejadas para o período de estágio foram completadas e algumas que não estavam no Plano de Estágio [Apêndice A] foram realizadas. O plano de estágio não contemplou todas as atividades, devido ao estagiário só vir estar consciente das mesmas após a elaboração do plano. Estas atividades resumem-se a elaboração de documentos, elaboração de treinamentos de tecnologias e estudo sobre as mesmas.

4.1 Metodologia de Desenvolvimento

No plano de estágio foi definido como metodologia de desenvolvimento o ADM que é a Metodologia de Desenvolvimento de software da Accenture, o ADM (Accenture Delivery Methods), que é um dos elementos chaves do ADS (Accenture Delivery Suite) da empresa. Este método apesar da possibilidade de se retornar a fases anteriores do processo de desenvolvimento, um dos conceitos chaves envolvidos na metodologia é o conceito de contenção de etapa. Todos os testes, validações e verificações são efetuados de maneira a reduzir ao máximo a possibilidade de que um problema gerado em uma etapa do processo passe para a etapa seguinte.

Para executar atividades de verificação e validação foi utilizado o V-Model. Este é um framework usado para estabelecer estágios bem definidos para orga-

nizar a verificação e a validação por todas as fases do desenvolvimento. Um diagrama das atividades pode ser observado na Figura 4.1.

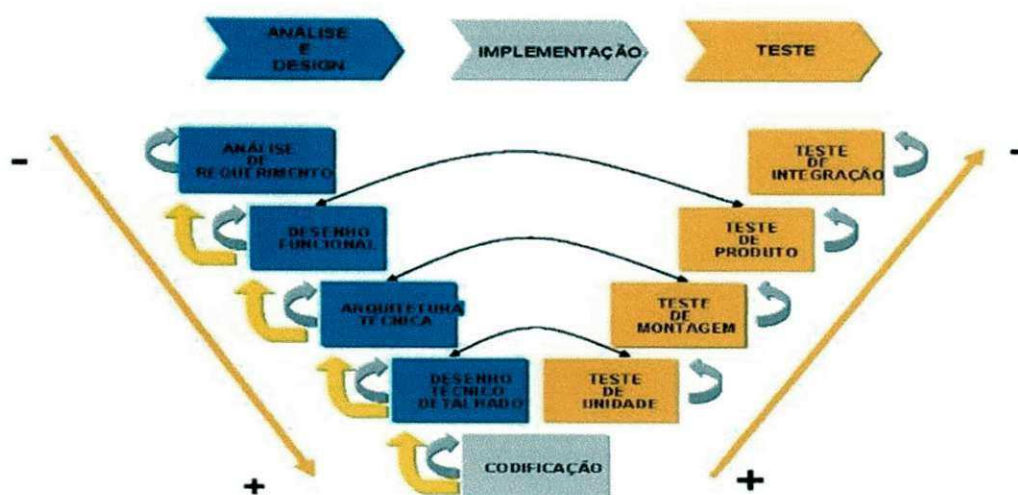


Figura 4.1: Framework V-Model

O desenvolvimento começa no lado esquerdo do V-Model com as atividades da análise e de projeto. A especificação é realizada de cima para baixo, tomando decisões e adicionando mais detalhe em cada estágio novo da especificação. Quando o projeto está completo, os processos da construção começam. Uma vez que a construção está completa, o produto move-se através das atividades de verificação, validação e teste no lado de mão direita do V. Durante os estágios iniciais dos testes, o foco está em componentes individuais. A medida que o teste progride, o foco passa para a funcionalidade e a realização do negócio.

No CoE a fase de análise de requerimento é realizada no cliente pela própria equipe do cliente.

Após receber o documento com a análise de requerimento, a equipe do CoE responsável pela área do requerimento elabora a Análise de Impacto.

O cliente aprovando o desenvolvimento, inicia-se a fase do desenho funcional (FD).

Ao término do FD, pode-se paralelizar a elaboração do desenho técnico (TD) e dos scripts de testes de produtos (PT).

Ao término do TD, pode-se paralelizar a elaboração dos scripts de Teste de Montagem (AT), a codificação, testes unitários e a elaboração do Manual de Operação e Change Form.

O teste de integração é especificado, caso necessário, junto ao documento referente ao teste de produto. Normalmente, o teste de integração é realizado no cliente pela própria equipe do cliente ou pela equipe da Accenture, quando é possível.

O CoE possui uma documentação padrão para cada fase do V-Model.

Todo artefato deve ser revisado, seguindo o procedimento de Peer Review. Para realizar a revisão, é necessário seguir um checklist de Peer Review, que contem todas as informações que devem ser avaliadas no documento.

Para um melhor entendimento das atividades do Framework V-Model, um fluxo de atividades pode ser observado nas Figuras 4.2, 4.3 e 4.4.

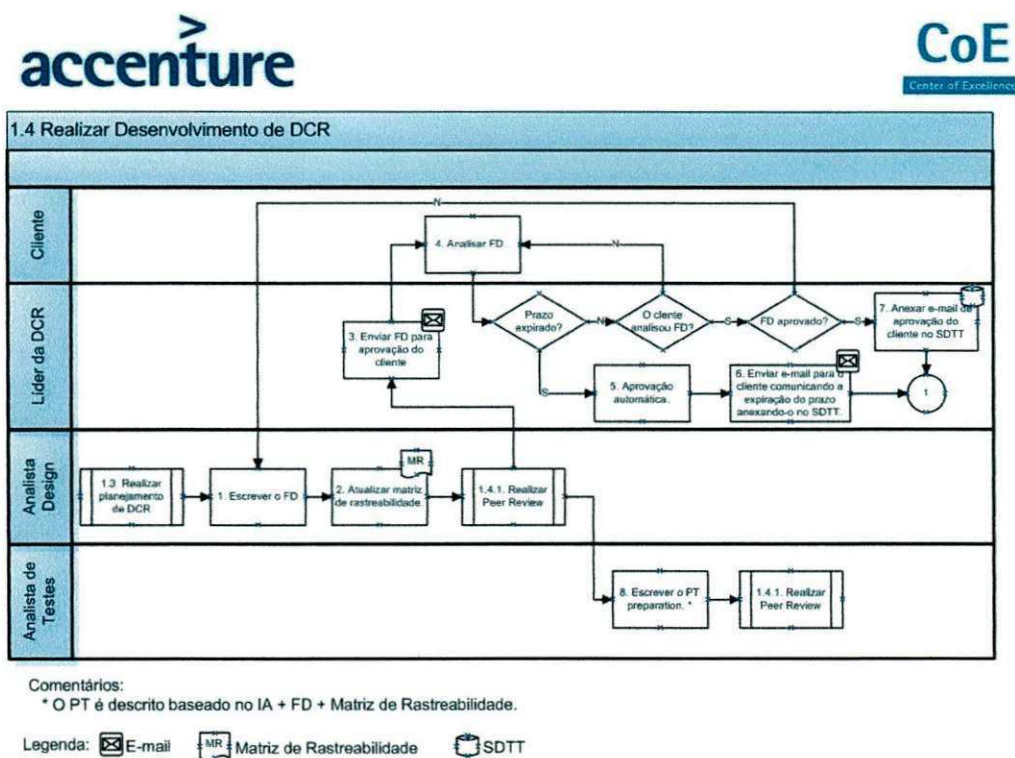
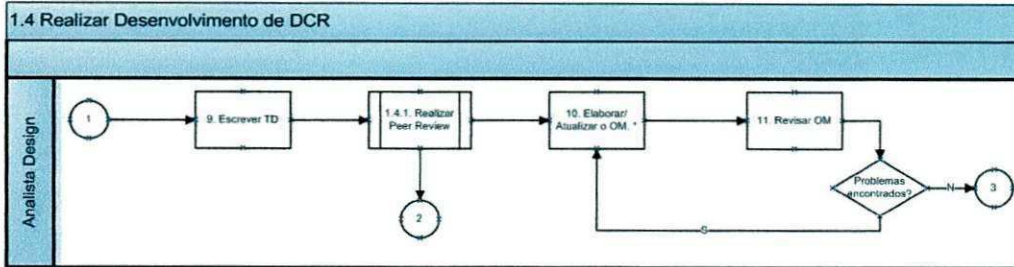


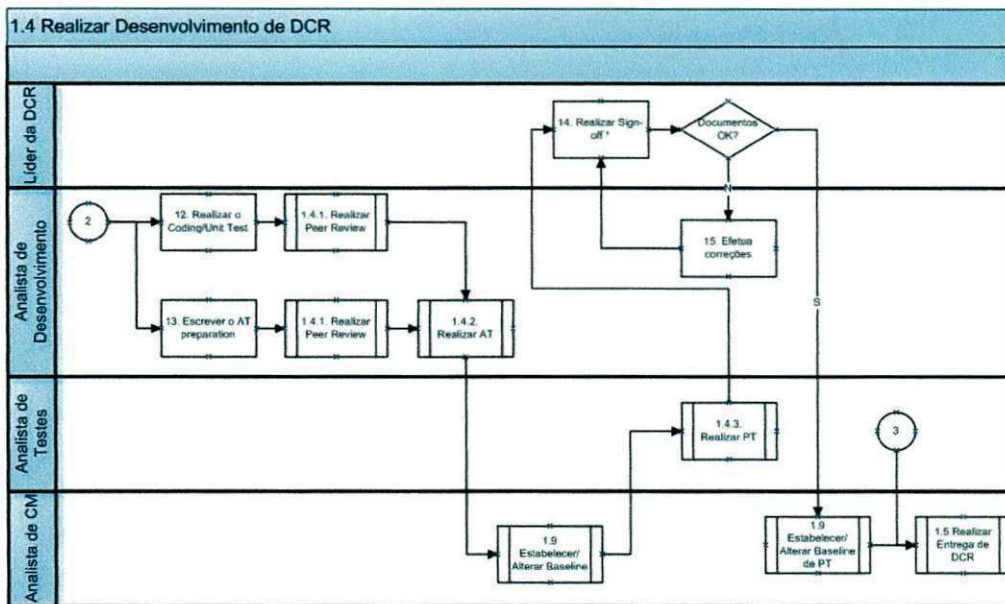
Figura 4.2: Diagrama de Atividades de Desenvolvimento - Parte 1.

Maiores detalhes sobre o processo de desenvolvimento podem ser observados no Anexo C.



Comentários:
 * Elaborar documentos específicos dos clientes se necessário (Cform, PF, OP).

Figura 4.3: Diagrama de Atividades de Desenvolvimento - Parte 2.



Comentários:
 * Os documentos revisados são: FD, TD, OM, AT preparation e PT preparation.

Figura 4.4: Diagrama de Atividades de Desenvolvimento - Parte 3

4.2 Dificuldades Encontradas Pela Equipe

Os módulos que foram desenvolvidos pela equipe, já existiam e precisava-se acrescentar funcionalidades aos mesmos. Todos os módulos dependiam de um outro módulo já existente, chamado Domain. Este foi desenvolvido por uma equipe de São Paulo. Devido ao domain não ter sido desenvolvido pela equipe de Campina Grande e a mesma era instruída a usar métodos do módulo, isto dificultou bastante o desenvolvimento, pois a equipe, inicialmente, começou a usar o domain sem conhecer o funcionamento do mesmo. Outra grande dificuldade enfrentada pela equipe foi que a mesma não teve tempo pra entender perfeitamente o funcionamento do sistema. O entendimento aconteceu paralelamente ao desenvolvimento, isto ocasionou atrasos nos prazos estabelecidos.

Os prazos para a realização das tarefas foram definidos por uma equipe da Accenture de São Paulo, estes prazos, na avaliação do coordenador do projeto, estavam subestimados até mesmo para desenvolvedores conhecedores do sistema. Esta subestimação de prazo atrapalhou bastante o desenvolvimento, visto que a equipe trabalhava sob pressão para cumprir prazos que nem uma equipe conhecedora do sistema era capaz de cumprir. Isto ocasionou várias reuniões, no intuito de discutir e resolver este problema. Em uma das reuniões, a equipe entrou em acordo com o líder do projeto e ficou determinado que cada desenvolvedor antes de iniciar sua tarefa, iria analisar e estimar o tempo necessário para conclusão da mesma. Após a estimativa do desenvolvedor o líder do projeto avaliava e dizia se o tempo foi bem estimado ou se a tarefa poderia ser concluída em menos tempo. Caso a segunda opção fosse a escolhida pelo líder, o mesmo argumentava o porque era possível terminar a tarefa em menos tempo. O desenvolvedor por sua vez poderia concordar ou discordar da afirmação do líder. Caso discordasse, teria que argumentar e convencer o líder que a estimativa feita pelo líder era insuficiente para concluir a tarefa.

4.3 Atividades Planejadas

As atividades previstas para o desenvolvimento do sistema, foram divididas entre a equipe de Campina Grande e a equipe de São Paulo. As atividades desenvolvidas pela equipe de Campina Grande foram:

- Codificação e Teste de Unidade (CUT)
- Teste de Montagem (AT)

- Teste de Produto (PT)
- Execução do Teste de Montagem (ATE)
- Execução do Teste de Produto (PTE)
- Peer Review

Como o projeto era relativamente grande, o estagiário teve a oportunidade de participar de todas as atividades alocadas para equipe de Campina Grande, o que normalmente não acontece. Geralmente um desenvolvedor é alocado para fazer duas ou no máximo três das atividades citadas anteriormente. As atividades desenvolvidas pelo estagiário podem ser observadas na seção 4.4.

O estudo de tecnologias, elaborações de documentos e treinamentos não tinha sido planejado, pois durante a elaboração do plano de estágio ainda não se tinha conhecimento das tecnologias que seriam utilizadas. No entanto, ao ficar ciente das tecnologias que seriam utilizadas, a equipe logo tomou iniciativa e decidiu que cada integrante iria estudar uma das tecnologias e elaborar um treinamento para a equipe. O treinamento consistiu de uma apresentação sobre a tecnologia e a criação de uma aplicação utilizando a mesma. Este processo ajudou bastante a equipe, pois todos aprenderam as tecnologias num curto espaço de tempo. O estagiário ficou responsável pela tecnologia Hibernate, o treinamento da mesma pode ser observado no Apêndice B.

4.4 Atividades Realizadas

Devido ao projeto ser consideravelmente extenso e um resumo das atividades já ter sido explicado nas seções anteriores, nesta seção, será detalhado apenas as atividades desenvolvidas pelo estagiário juntamente com as maiores dificuldades da atividade e os benefícios que a mesma trouxe.

4.4.1 Novas Tecnologias

Como foi citado anteriormente, a equipe se dividiu e cada integrante elaborou o treinamentos referente a tecnologia que lhe coube a responsabilidade. Foi elaborada uma apresentação de Hibernate [HIBERNATE] conforme presente no Apêndice C. O treinamento consistiu em ensinar como realizar mapeamentos do Hibernate usando xml.

Dificuldades

Maior dificuldade durante o estudo da tecnologia foi realizar os relacionamentos das classes em xml, maneira esta exigida pelo projeto devido a usar o Hibernate 3.

Benefícios

A elaboração do treinamento trouxe benefícios importantes para o estagiário, pois o mesmo aprendeu mais uma maneira de realizar o mapeamento entre as classes javas e as tabelas do banco, utilizando Hibernate.

4.4.2 Coding Unit Test (CUT)

Esta tarefa consistiu em implementar as funcionalidades do Loader de Arrecadação.

O novo módulo foi responsável por realizar a carga das informações de arrecadação e inadimplência de serviços diferenciados. Este, irá processar os arquivos recebidos pelo RMCA. Estes arquivos possuíam informações de faturas arrecadadas, inadimplentes ou recuperadas de inadimplência.

Durante esta tarefa foram implementadas 6 classes e 32 métodos, não contabilizando os métodos *gets* e *setters* dos atributos contidos nas classes.

A tarefa foi dividida da seguinte maneira:

- Processamento principal.
- Leitura do arquivo e parse dos registros.
- Tratamento para arrecadação.
- Tratamento para inadimplência.
- Atualização de Peer Review.

Dificuldades

Esta foi umas das tarefas mais difíceis durante o desenvolvimento do projeto, apesar do estagiário possuir experiência na atividade. A maior dificuldade deu-se devido ao estagiário não possuir conhecimento suficiente de todo o sistema que seria modificado.

O desconhecimento do sistema se deu devido as dificuldades citadas na seção 4.2.

Benefícios

Com esta tarefa conclui-se que uma alteração de um sistema, deve-se começar, se, e somente se, os desenvolvedores possuam conhecimento suficiente do sistema. Apesar deste conceito já ser de conhecimento do estagiário, o mesmo não pode ser aplicado devido à ordens superiores.

4.4.3 Peer Review

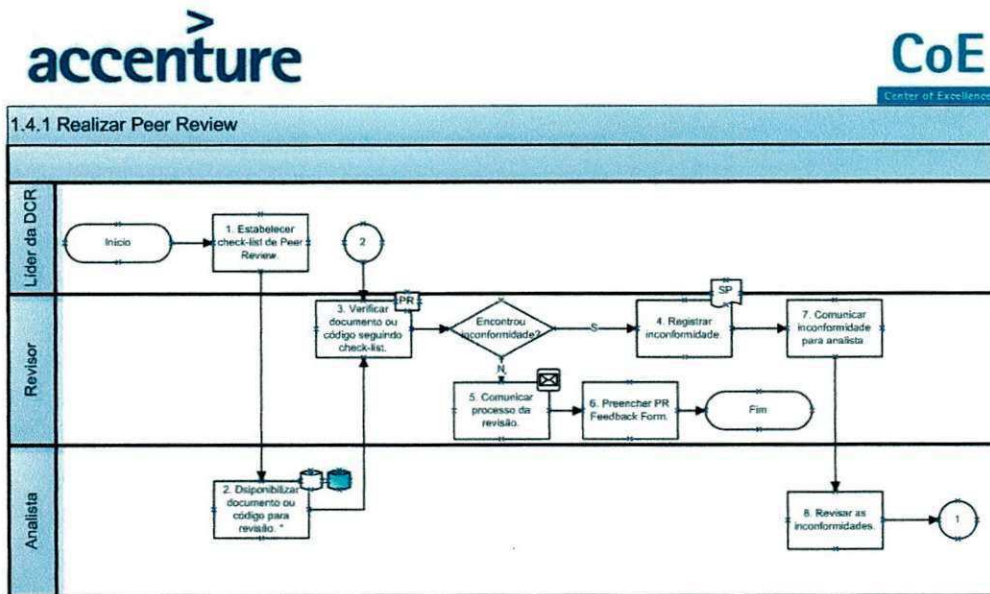
Esta atividade consistiu em revisar todos artefatos. Para realizar a revisão, foi necessário seguir um checklist de Peer Review, que contém todas as informações que devem ser avaliadas no documento. Um diagrama do fluxo de atividades do Peer Review pode ser observado nas Figuras 4.5 e 4.6.

Dificuldades

Durante esta tarefa não foi encontrada dificuldades.

Benefícios

Devido esta atividade fazer parte do processo de desenvolvimento da Accenture, é bastante importante, que um funcionário da mesma, saiba como funciona e como realizá-la.



Comentários:
 * Disponibilizar IA/CP no SDTT ou outros documentos no Controlador de Versões.

Legenda:
 SDTT Controlador de Versões Checklist de Peer Review Sharepoint E-mail

Figura 4.5: Diagrama de Atividades do Peer Review Parte 1.

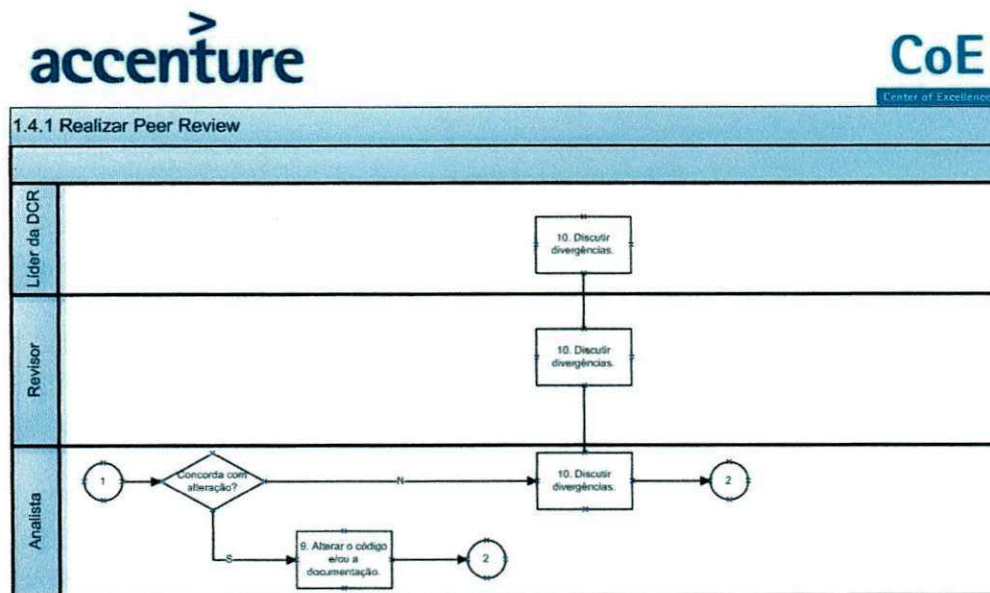


Figura 4.6: Diagrama de Atividades do Peer Review Parte 2.

4.4.4 Assembly Test Preparation (ATP)

Após a tarefa de CUT, deu-se início a criação do AT. Lembrado que o AT elaborado foi de um módulo que não estava sendo desenvolvido pela equipe. Os testes de AT verificam se a inter-relação entre as unidades e módulos do sistema e se o fluxo de informação no sistema está ocorrendo como previsto no Desenho Técnico. É neste teste que se realiza o teste de montagem (comunicação entre as várias partes do módulo).

Dificuldades

Não foram encontradas grandes dificuldades na elaboração deste documento, apesar de ser uma tarefa nova para o estagiário, o desenvolvimento foi dado de forma gradual. O início da tarefa foi a parte mais crítica, devido o estagiário nunca ter elaborado este tipo de documento. A tarefa se desenvolveu da seguinte maneira: primeiramente, o estagiário elaborou uma pequena parte do documento e este foi apresentado ao líder do projeto. Em seguida o líder informou o que estava correto e o que precisaria ser modificado e como modificar, para que o documento estivesse nos padrões da Accenture. As modificações foram efetuadas e o documento foi novamente exibido ao líder. Este processo continuou até que o líder informou ao estagiário que ele seguisse aquela linha de desenvolvimento.

Benefícios

Devido este documento fazer parte do processo de desenvolvimento da Accenture, é bastante importante, que um funcionário da mesma, saiba como funciona e como elaborar este documento.

4.4.5 Assembly Test Execution (ATE)

Nesta fase, é preciso realizar todas as configurações solicitadas no documento de AT. Após este passo, é executado sequencialmente os passos de teste descritos no documento. Para cada passo de teste existem cenários a serem montados. Caso durante a execução do passo seja encontrado uma inconformidade, a mesma deverá ser cadastrada no portal da DCR para que um outro analista possa estar corrigindo-a.

A execução do ATE é finalizada quando todos os passos de teste forem exe-

cutados com sucesso. Após isto, o Analista de desenvolvimento deverá dar um check-in no documento e informar o líder a conclusão da atividade.

Um diagrama de atividades do ATE pode ser observado na Figura 4.7

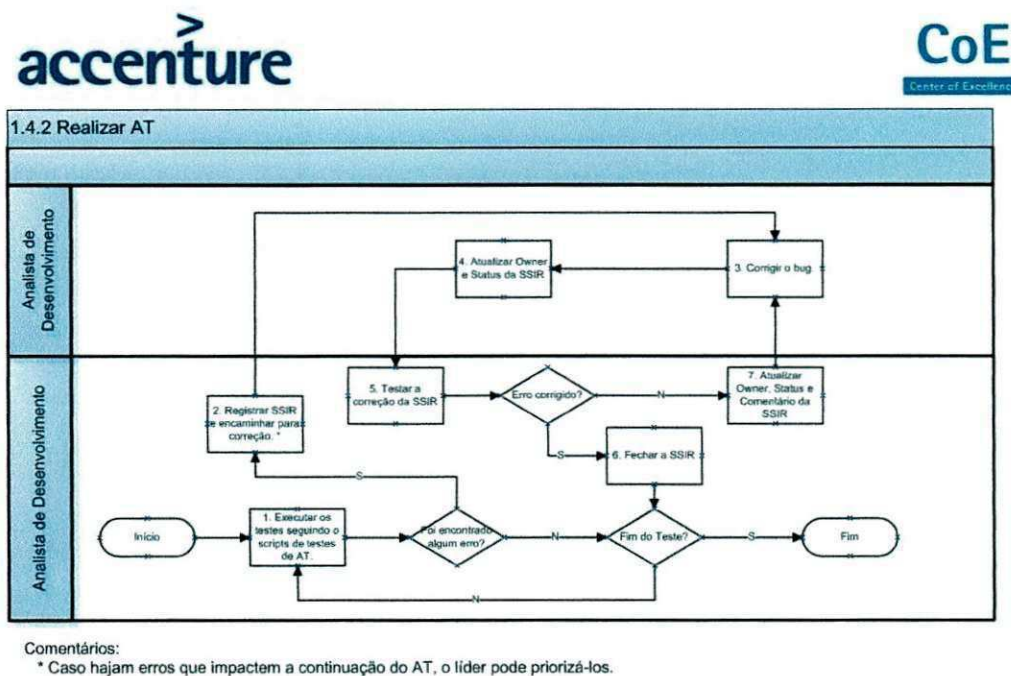


Figura 4.7: Diagrama de Atividades do Assembly Test Execution.

Dificuldades

A maior dificuldade encontrada nesta fase foi executar os testes. O módulo que estava sendo testado dependia de um outro módulo, este, por sua vez, possuía bastante inconsistência referente ao banco de dados utilizado. As maiores inconsistências foram: nomes de tabela e atributos diferentes, na maioria das vezes a diferença era de letras. Isto ocasionou bastante atraso na execução da tarefa, visto que era preciso ir até o módulo que continha os erros e conferir o nome dos atributos e/ou tabelas com os nomes dos atributos e/ou tabelas do banco.

Lembrando que a correção destas inconsistências não era de responsabilidade da equipe, mas mesmo assim, as mesmas foram feitas.

Benefícios

Com esta atividade aprendeu-se que se uma atividade não é de sua responsabilidade, passe-a para quem é o responsável.

Artefatos

A Accenture não permite divulgação de qualquer conteúdo produzido em qualquer projeto. Nas Figuras 4.8 e 4.9 é possível observar a alocação do estagiário, juntamente com as horas gastas em cada tarefa.

View my tasks

Task Name	Actual	Remain	Start	Finish	Resource Name	Work
DCR TIM1289 - GVAS	197h	88.6h	3/24/2009	5/6/2009		285.6h
DCR TIM1289 - GVAS -	197h	88.6h	3/24/2009	5/6/2009		285.6h
Loader/Extrator Faturados	42.5h	0h	3/24/2009	4/29/2009		42.5h
DSV	42.5h	0h	3/24/2009	4/29/2009		42.5h
Kick-off	0.5h	0h	3/24/2009	3/24/2009	Tiago de Lima Wanderley	0.5h
Assembly Test Execution	36h	0h	4/7/2009	4/14/2009		36h
ATE - Loader Faturados	36h	0h	4/7/2009	4/14/2009	Tiago de Lima Wa	36h
Product Test Execution	6h	0h	4/22/2009	4/29/2009		6h
PTE - Bug Fixing	6h	0h	4/22/2009	4/29/2009	Tiago de Lima Wa	6h
Loader de Serviços/ Loader Arrecadação	106h	27.4h	3/24/2009	4/16/2009		133.4h
DSV	106h	27.4h	3/24/2009	4/16/2009		133.4h
Coding & Unit Test	106h	5h	3/24/2009	4/6/2009		111h
CUT - Loader Arrec. - Processamento principal	13h	0h	3/24/2009	4/2/2009	Tiago de Lima Wa	13h
CUT - Loader Arrec. - Leitura do arquivo e parse dos registros	20h	0h	3/24/2009	4/3/2009	Tiago de Lima Wa	20h
CUT - Loader Arrec. - Tratamento para arrecadação	32h	0h	3/25/2009	4/2/2009	Tiago de Lima Wa	32h
CUT - Loader Arrec. - Tratamento para inadimplência	30h	0h	3/27/2009	4/3/2009	Tiago de Lima Wa	30h
CUT - Atualização de Peer Review	11h	0h	3/29/2009	4/6/2009	Tiago de Lima Wa	11h
CUT - Loader Arrec. - Inserção/Atualização das tabelas da base	0h	5h	3/31/2009	4/1/2009	Tiago de Lima Wa	5h
Assembly Test Execution	0h	22.4h	4/6/2009	4/14/2009		22.4h
ATE - Loader Serviços	0h	22.4h	4/6/2009	4/14/2009	Tiago de Lima Wa	22.4h
Product Test Execution	0h	0h	4/16/2009	4/16/2009		0h
PTE - Bug Fixing - Loader Serviços	0h	0h	4/16/2009	4/16/2009	Tiago de Lima Wa	0h
GVAS Online	48.5h	61.2h	4/14/2009	5/6/2009		109.7h
DSV	48.5h	61.2h	4/14/2009	5/6/2009		109.7h

Figura 4.8: Project Parte 1.

View my tasks

View Options Filter, Group, Search						
Reject Insert Notes Link Risks Link Issues Link Documents 5/28/2009-7/4/2009						
Task Name	Actual	Remain	Start	Finish	Resource Name	Work
GVAS Online	48.5h	61.2h	4/14/2009	5/6/2009		109.7h
DSV	48.5h	61.2h	4/14/2009	5/6/2009		109.7h
Assembly Test Preparation	40h	4.8h	4/14/2009	4/30/2009		44.8h
ATP - Online 01	18h	0h	4/14/2009	4/23/2009	Tiago de Lima Wa	18h
ATP - Online 02	22h	0h	4/17/2009	4/22/2009	Tiago de Lima Wa	22h
ATP - Atualização de Peer Review	0h	4.8h	4/29/2009	4/30/2009	Tiago de Lima Wa	4.8h
Coding & Unit Test	0h	56.4h	4/15/2009	5/4/2009		56.4h
CUT - Menus, Perfis e Funcionalidades	0h	10h	4/15/2009	4/17/2009	Tiago de Lima Wa	10h
CUT - Cadastro de serviço diferenciado	0h	15h	4/17/2009	4/20/2009	Tiago de Lima Wa	15h
CUT - Cadastro de contratos	0h	14.6h	4/20/2009	4/23/2009	Tiago de Lima Wa	14.6h
CUT - Cadastro de fornecedores (contatos)	0h	4h	4/23/2009	4/24/2009	Tiago de Lima Wa	4h
CUT - Atualização de Peer Review	0h	12.8h	4/29/2009	5/4/2009	Tiago de Lima Wa	12.8h
Assembly Test Execution	8.5h	0h	5/4/2009	5/6/2009		8.5h
ATE - Online - Cadastro de Serviço/Fornecedor	8.5h	0h	5/4/2009	5/6/2009	Tiago de Lima Wa	8.5h

Figura 4.9: Project Parte 2.

Capítulo 5

Resultados esperados x Resultados obtidos

Os resultados esperados, juntamente com os obtidos, podem ser observados nas Tabelas 5.1 e 5.2.

Tabela 5.1: Atividades de Cadastro realizadas e seus respectivos status

Novos Cadastros	Módulo Batch	Status Final	Total de Bugs
Cadastro de Serviços Diferenciados	Report_Cob_Terc	OK	2
Cadastro de Fornecedor	Report_Cob_Terc	OK	3
Cadastro de Contratos	Report_Cob_Terc	OK	2

Tabela 5.2: Telas criadas e seus respectivos status

Telas de Relatórios	Módulo	Status	Total de Bugs
Relatório de Repasse	Report_Cob_Terc	OK	3
Relatório de Repasse Detalhado	Report_Cob_Terc	OK	2
Relatório de Status Detalhado	Report_Cob_Terc	OK	5
Relatório Detalhado de Receitas Geradas	Report_Cob_Terc	OK	1

5.1 Módulos criados

Os módulos que foram criados com os seus respectivos status podem ser observados na Tabela 5.3.

Tabela 5.3: Módulos criados e seus respectivos status

Nome do Módulo	Status Final	Passos	Total de Bugs
Loader Serviços BSCS	OK	28	13
Extrator de faturamento de serviços	OK	24	3
Loader de faturamento de serviços	OK	33	15
Loader Arrecadação RMCA	OK	44	12

5.2 Resultados Gerais Obtidos

- O projeto GVAS foi desenvolvido dentro do prazo previamente estimado (250 mdays = 2000 horas = 84 dias);
- Todas as funcionalidades exigidas foram completamente desenvolvidas;
- O projeto entrou em produção no ambiente da PHONEMAX;

Em termos administrativos, o projeto finalizou com déficit, caracterizando um estouro de budget, conseqüentemente uma penalização na lucratividade esperada. Tal perda foi devidamente justificada devido aos problemas encontrados em módulos de reaproveitamento, bem como da sumarização do projeto em apenas um integrante (Thiago Pimentel - Líder de SP).

5.3 Resultados esperados do estagiário

Os resultados esperados do estagiário podem ser observados na Tabela 5.4.

Tabela 5.4: Resultados esperados do estagiário

Módulo	Atividade	Período
Loader/Extrator Faturados	ATE/PTE	7/4 à 14/4 (36 Horas)
Loader Arrecadação	CUT	22/4 à 29/04 (111 Horas)
GVAS Online	3 ATP's	14/4 à 30/4 (44,8 Horas)
GVAS Online	ATE	4/5 à 6/5 (8,5 Horas)
Total de Horas		200,3 Horas

5.4 Resultados obtidos do estagiário

Todas as atividades foram finalizadas.

No CUT realizado pelo estagiário foram encontrados 12 bugs de um total de 44 passos.

5.5 Desempenho do Estagiário

O texto a seguir foi elaborado pelo supervisor técnico, Francisco de Barros Neto.

“O estagiário mostrou-se capaz de desenvolver as atividades para as quais foi alocado. Trabalhou com total dedicação inclusive, quando necessário, nos finais de semana pra finalizar atividades não concluídas durante a semana. Mostrou espírito de equipe e soube lidar com a experiência de um projeto longo e delicado. Em algumas atividades, o estagiário não conseguiu realizá-las no tempo estimado, no entanto, este atraso sempre foi justificável por motivos extra fábrica de Campina Grande.”

Capítulo 6

Considerações Finais

O desenvolvimento de um sistema como o GVAS é realmente uma tarefa complexa e extensa. É de suma importância, que os profissionais ou estagiários envolvidos em tarefas semelhantes busquem metodologias de desenvolvimento que visem a modularização do sistema.

A oportunidade de colocar em prática teorias compreendidas em ambiente acadêmico é realmente muito gratificante e fundamental na formação profissional de qualquer estudante. Tive o prazer de aprofundar meus conhecimentos em Programação, Banco de Dados, Sistemas de Informação e Engenharia de Software, além de ter noção de como é trabalhar em uma empresa de grande porte com processos de desenvolvimento consolidados e em andamento.

Desta forma, este estágio possibilitou a execução dos seguintes pontos: utilizar técnicas de Teste de Software para programação Orientada a Objetos, com intuito de prover à aplicação uma maior robustez, eficiência, escalabilidade e corretude; praticar a utilização de “frameworks” da área de desenvolvimento de componentes; praticar a habilidade de adquirir novos conhecimentos necessários para a solução do problema e assimilar novas tecnologias.

Desta forma, este estágio realizado foi de grande valia e possibilitou a solução de problemas como validação de software, verificação da qualidade de software, utilizando as mais diversas tecnologias e metodologias de desenvolvimento de software do mundo computacional moderno.

O ensino da Universidade onde o estagiário estuda é muito bom, mas existem falhas na aplicação de metodologias adotadas em algumas disciplinas. A metodologia é correta, o problema é como ela é aplicada e cobrada. Como exemplo pode ser citada a disciplina de Laboratório de Engenharia de Software (LES),

existem prazos determinados para a entrega e apresentação de releases. No entanto, a avaliação do professor quanto ao que foi feito é bastante superficial, não comprovando se a equipe fez o que realmente foi pedido ou não. Com isto, a disciplina perde um dos seus objetivos, que é simular a cobrança e cumprimento de prazos estabelecidos, o que normalmente é visto em empresas. Para resolver tal situação, era preciso que o professor a cada release, pedisse aos alunos que mostrassem a aplicação contemplando a implementação das funcionalidades planejadas para aquela release.

Referências Bibliográficas

[Braz]BRAZ, C. C. M. *Principais padrões J2EE para a construção de aplicações não distribuídas*. Disponível em: http://www.guj.com.br/content/articles/patterns/padroes_guj.pdf. Acesso em 01 de Julho de 2009.

[Fechine 2009]FECHINE, J. M. 2009. Disponível em: <http://www.dsc.ufcg.edu.br//joseana>. Acesso em 01 de Julho de 2009.

[Fechine 2009]FECHINE, J. M. *Estágio Integrado*. 2009. Disponível em: <http://www.dsc.ufcg.edu.br//joseana/EI-20091.html>. Acesso em 01 de Julho de 2009.

[HIBERNATE]HIBERNATE. Disponível em: <http://www.hibernate.org/>. Acesso em 01 de Julho de 2009.

[Tiles]TILES. Disponível em: http://struts.apache.org/userGuide/dev_tiles.html. Acesso em 01 de Julho de 2009.

[WIKIPEDIA]WIKIPEDIA.
Disponível em: http://pt.wikipedia.org/wiki/BEA_Weblogic_Server. Acesso em 01 de Julho de 2009.

[WIKIPEDIA]WIKIPEDIA.
Disponível em: http://pt.wikipedia.org/wiki/Spring_Framework. Acesso em 01 de Julho de 2009.

Apêndice A

Plano De Estágio



UFPG – UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CEEI – CENTRO DE ENG^a ELÉTRICA E INFORMÁTICA
DSC – DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

Plano de Estágio Integrado

*Empresa de consultoria de gestão, serviços de tecnologia e
outsourcing.*

Accenture - Campina Grande

Tiago de Lima Wanderley

Mat.: 20521157

Orientadora: Prof. Joseluze de Farias Cunha

Março 2009

Informações Pessoais

Nome: Tiago de Lima Wanderley

Endereço: Rua Rodrigues Alves, 1440 – CEP:58.428-795

Bela Vista, Campina Grande, PB - Brasil

Email: tiagowanderleyufcg@gmail.com

Ambiente do Estágio

A execução do estágio será realizada na Fábrica da Accenture de Campina Grande[1] situada à Rua Maria Vieira César, 230 – Alto Branco.

A cultura da empresa utiliza-se da organização dos desenvolvedores em pequenos grupos chamados *células*. Estas foram divididas de forma que se tenha em cada uma, uma equipe responsável pelos procedimentos de análise, desenvolvimento, testes de software e gerenciamento das atividades em curso.

Supervisão

Supervisor Acadêmico

Nome: Joseluze de Farias Cunha

Endereço: Departamento de Sistemas e Computação

Universidade Federal de Campina Grande

Avenida Aprígio Veloso, 882 – CEP: 58.109-970

Bodocongó, Campina Grande, PB – Brasil.

Email: joseluze@dsc.ufcg.edu.br

Supervisor Técnico

Nome: Francisco Manoel de Barros Neto

Endereço: Rua Sinhazinha Celino, 65 – CEP: 58105-613

Distrito Industrial, Campina Grande, PB - Brasil

Email: kiko.ufcg@gmail.com

Objetivos do Estágio

Esse plano de estágio tem como objetivo fornecer informações sobre as atividades que serão desempenhadas pelo aluno Tiago de Lima Wanderley, do curso de Ciência da Computação da UFCG, durante seu período de estágio na Fábrica da Accenture em Campina Grande.

Esse período de estágio na empresa será desenvolvido juntamente com mais quatro pessoas. O objetivo é modificar um sistema para atender as novas exigências impostas pela empresa de telefonia.

O projeto surgiu quando uma empresa de telecomunicação do Brasil identificou a necessidade de inclusão de novas funcionalidades no sistema **GVAS**, sistema este, responsável por repasse e ressarcimento, configurações de contratos e serviços, aplicáveis a "**serviços diferenciados**", **independentemente de tráfego de dados**.

O GVAS atual, funciona de maneira parecida, só que ao contrário do que vai ser implementado, este, apenas contabiliza serviços da própria empresa como por exemplo, papéis de parede, *ringtones* e etc.

Dentre as tecnologias envolvidas, temos: Servidor Weblogic 8.1, Struts 1.2, Hibernate, Spring, Tiles, Java.

Resumo do Problema

Cada vez mais, o celular deixa de ser uma simples ferramenta de comunicação por voz, sendo utilizado para o envio e recebimento de *emails*, *download* de vídeos, música, imagens, recebimento de notícias, etc.

Como grande parte destes novos serviços é provida por plataformas de rede gerenciadas por outras empresas, a Empresa X deve repassar a estas empresas parte do valor arrecadado com a utilização destes serviços por seus usuários.

O sistema GVAS é responsável pelo gerenciamento financeiro e de repasse aos provedores pela prestação de serviços, mediante as regras definidas pelo contrato estabelecido entre a Operadora de telefonia e o fornecedor.

Neste sistema deverão ser registrados todos os serviços utilizados por clientes da Empresa X que tenham sido providos por plataformas gerenciadas por empresas terceiras, acompanhando também o ciclo de vida destes serviços (faturamento, arrecadação, contestação, rejeição etc.).

O GVAS é composto por um módulo Online, através do qual podem ser cadastrados os fornecedores, serviços e contratos, configurados os tipos de serviços, plataformas e usuários e gerados relatórios gerenciais, e um módulo batch, responsável por registrar na base de dados a utilização destes serviços por clientes da Empresa X.

A empresa de telefonia identificou a necessidade de inclusão de novas funcionalidades no **GVAS**, gerando impacto nos módulos *Batch*, *Relatórios* e *OnLine*. Tais funcionalidades geraram a necessidade de desenvolver regras de

repass e ressarcimento, configurações de contratos e serviços, aplicáveis a "serviços diferenciados", independentemente de tráfego de dados.

Proposta de Solução

Para atender a solicitação da Empresa de telefonia, o sistema GVAS será alterado para contemplar o repasse para os novos serviços diferenciados.

Resumidamente, os seguintes processos serão criados/alterados:

- GVAS *Online*;
- GVAS *Batch*;
- GVAS Relatórios.

Metodologia

ADM

O CoE utiliza a Metodologia de Desenvolvimento de software da Accenture, o ADM (Accenture Delivery Methods), que é um dos elementos chaves do ADS (Accenture Delivery Suite) da empresa.

Os métodos do ADM são consistentes com o Capability Maturity Model (CMM), garantindo a cobertura de todos os Key Process Áreas (KPA) nos níveis II e III do CMM.

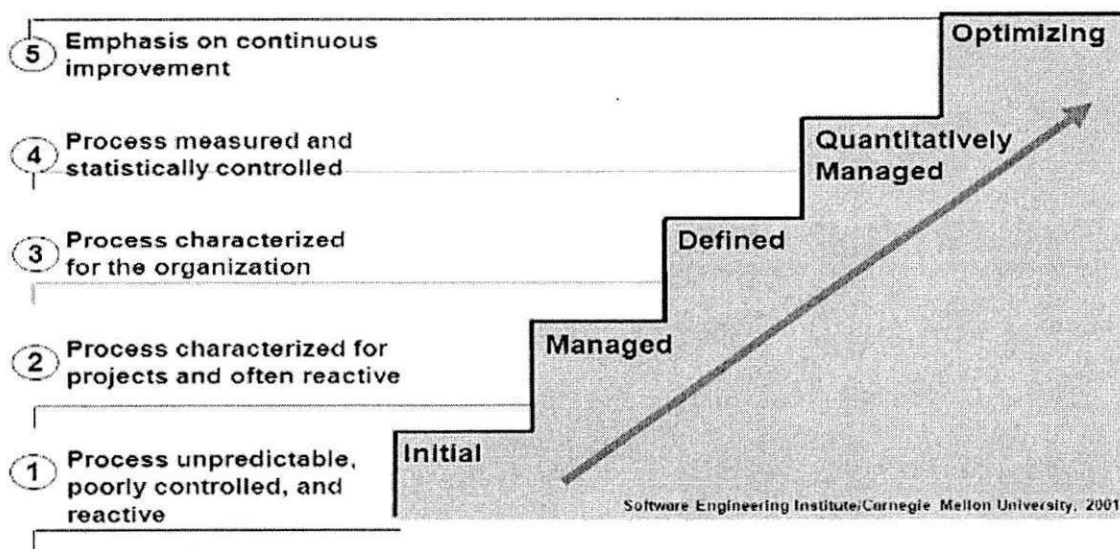
Apesar da possibilidade de se retornar a fases anteriores do processo de desenvolvimento, um dos conceitos chaves envolvidos na metodologia é o conceito de contenção de etapa. Todos os testes, validações e verificações são efetuados de maneira a reduzir ao máximo a possibilidade de que um problema gerado em uma etapa do processo passe para a etapa seguinte.

CMMi

O Modelo Integrado de Maturidade da Capabilidade de Software (CMMI) fornece às organizações de software um guia de como obter controle em seus processos para desenvolver e manter software e como evoluir em direção a uma cultura de engenharia de software e excelência de gestão.

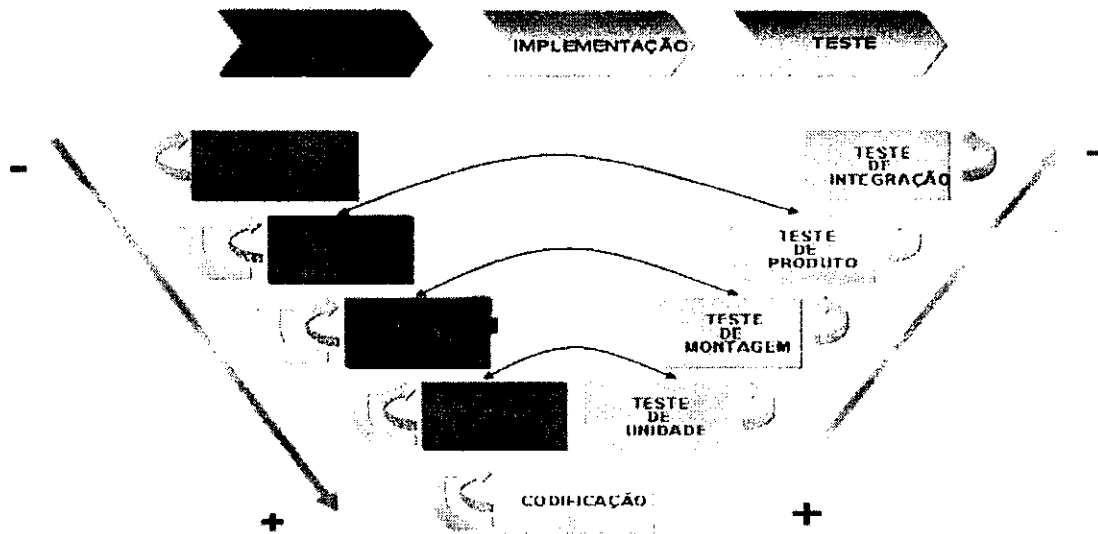
A melhoria contínua de processo é baseada em muitas etapas evolutivas pequenas ao invés de fundamentar-se em inovações revolucionárias . O CMMI fornece uma estrutura para organizar essas etapas evolutivas em cinco níveis de maturidade que estabelecem fundamentos sucessivos para a contínua melhoria do processo.

Esses cinco níveis de maturidade definem uma escala ordinal para medir a maturidade de um processo de software da organização e para avaliar a sua capacidade de processo de software. Os níveis também ajudam uma organização a priorizar seus esforços de melhorias.



V-Model

O projeto CoE usa o V-Model para executar atividades de verificação e validação. O V-Model é um framework usado para estabelecer estágios bem definidos para organizar a verificação e a validação por todas as fases do desenvolvimento.



O desenvolvimento começa no lado esquerdo do V-Model com as atividades da análise e de projeto. A especificação é realizada de cima para baixo, tomando decisões e adicionando mais detalhe em cada estágio novo da especificação. Quando o projeto está completo, os processos da construção começam. Uma vez que a construção está completa, o produto move-se através das atividades de verificação, validação e teste no lado de mão direita do V. Durante os estágios iniciais dos testes, o foco está em componentes individuais. A medida que o teste progride, o foco passa para a funcionalidade e a realização do negócio.

O processo de verificação e validação é uma tentativa de tratar os problemas o mais cedo possível no ciclo de vida do desenvolvimento e de assegurar que as especificações estejam completas, corretas e adequadas aos padrões. O processo de testar assegura que as especificações tenham sido corretamente implementadas e que a solução atenda as exigências do negócio e de desempenho.

Metodologia Desenvolvimento CoE

No CoE a fase de análise de requerimento é realizada no cliente pela própria equipe do cliente.

Após receber o documento com a análise de requerimento, a equipe do CoE responsável pela área do requerimento elabora a Análise de Impacto.

O cliente aprovando o desenvolvimento inicia-se a fase do desenho funcional (FD).

Ao término do FD, pode-se paralelizar a elaboração do desenho técnico (TD) e dos scripts de testes de produtos (PT).

Ao término do TD, pode-se paralelizar a elaboração dos scripts de Teste de Montagem (AT), a codificação, testes unitários e a elaboração do Manual de Operação e Change Form.

O teste de integração é especificado, caso necessário, junto ao documento referente ao teste de produto. Normalmente, o teste de integração é realizado no cliente pela própria equipe do cliente ou pela equipe da Accenture, quando é possível.

O CoE possui uma documentação padrão para cada fase do V-Model.

Todo artefato deve ser revisado, seguindo o procedimento de Peer Review. Para realizar a revisão, é necessário seguir um checklist de Peer Review, que contem todas as informações que dever ser avaliadas no documento.

Benefícios

Ao usar o V-Model durante todo o projeto, pode-se obter principalmente os seguintes benefícios:

- Melhoria na qualidade e confiança
- Redução da quantidade de re-trabalho;
- Redução do custo de correção do problema;
- Entrega no tempo acordado.

QPI

O QPI é um programa que visa garantir que todos os projetos seguem o processo do ADM com nível mínimo de maturidade 3 do CMMi.

A execução das práticas do QPI são essenciais para um projeto bem sucedido. Estas práticas aplicam-se tanto quando usa-se o Accenture Delivery Methods quanto para os métodos do cliente.

Periodicamente é realizado uma auditoria no projeto, fazendo parte do processo a coleta de evidências e entrevistas com membros do CoE.

A equipe de Qualidade é responsável por preparar o projeto para a auditoria, ou seja, mantém os processos sempre vivos e dinâmicos, além dos planos atualizados. Garante que os membros do projeto têm conhecimento do processo como um todo através de auditorias internas.

Os planos seguidos pelo QPI e que devem ser do conhecimento de todos são (disponíveis no Portal da Qualidade):

Business Continuity Approach

É o Plano de Contingência, onde descrevemos o que deve ser feito em caso de uma tragédia/desastre, ou seja, em caso de acidentes no projeto e/ou na Accenture que cause a parada no desenvolvimento para o cliente.

Communication Plan

No Plano de Comunicação descrevemos todos os tipos de comunicação realizadas no projeto. Exemplos: emails enviados, PortalCoE, reuniões etc.

Configuration Management Plan

A Gerência de Configuração (CM) permite o gerenciamento controlado e repetível dos componentes da Tecnologia de Informação (IT), pois envolve todas as fases do desenvolvimento e manutenção. CM implementa um processo no qual os integrantes do projeto e todos os envolvidos identificam, comunicam, implementam, documentam e gerenciam as mudanças. Quando devidamente implementado, CM mantém a integridade de todos os itens que forem sendo colocados sob seu controle.

Knowledge Management Plan

O Plano de Gestão do Conhecimento tem o propósito de estabelecer como compartilhar, armazenar, gerenciar, organizar os conhecimentos do Projeto CoE. Além disso, determinar qual conteúdo deve ser disponibilizado para uso dos recursos humanos do projeto.

Quality Management Plan

O gerenciamento da qualidade é a abordagem sistemática que consiste em processos, procedimentos, técnicas e padrões para garantir:

- A qualidade dos produtos gerados pelo projeto;
- A aplicação de melhores práticas e técnicas que minimizam os riscos com prazos, budget e qualidade;
- O entendimento das expectativas dos stakeholders, com a clara definição da abordagem para atendimento dessas expectativas.

Risk Management Plan

O objetivo do Plano de Gerenciamento de Riscos é orientar a necessidade de gerenciar e monitorar riscos para evitar atrasos, aumento de custo, problemas de desempenho, impactos ambientais adversos ou outras conseqüências indesejáveis.

Unit Operating Model

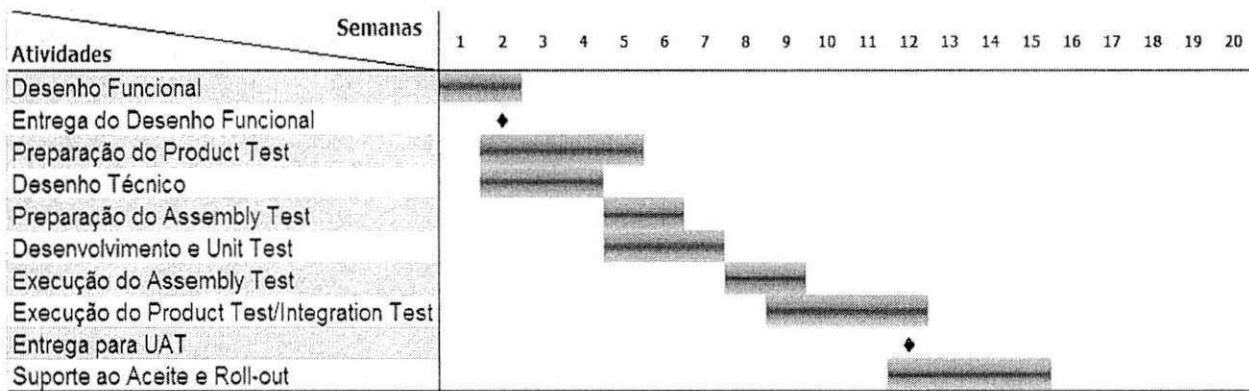
A leitura do Unit Operating Model do projeto CoE é obrigatória para todos os seus membros. O Unit Operating Model é o guia geral da unidade. Ele descreve o overview da unidade, os processos padrões, e a estrutura organizacional.

O Unit Operating Model deve ser visualizado por todos os membros e deve ser uma fonte contínua de informações para todos os membros. Ele fornece um overview dos processos da unidade como também informações específicas sobre outros documentos.

Unit Measurement Plan

O propósito do Plano de Métricas do Projeto CoE é definir as métricas que irão dar suporte à gerência para tomar decisões baseadas em informações para promover qualidade, produtividade e melhoria de processo. Este plano assegura que as métricas definidas são alinhadas aos objetivos do negócio e do programa, e que as métricas são implantadas em uma abordagem organizada e planejada.

Atividades Planejadas & Cronograma



*** A depender da demanda do CoE e da priorização desta DCR por parte da Empresa de Telecomunicações.**

Referências Bibliográficas

[1] Site da Fábrica Accenture. <http://www.accenture.com.br>.

Aprovação

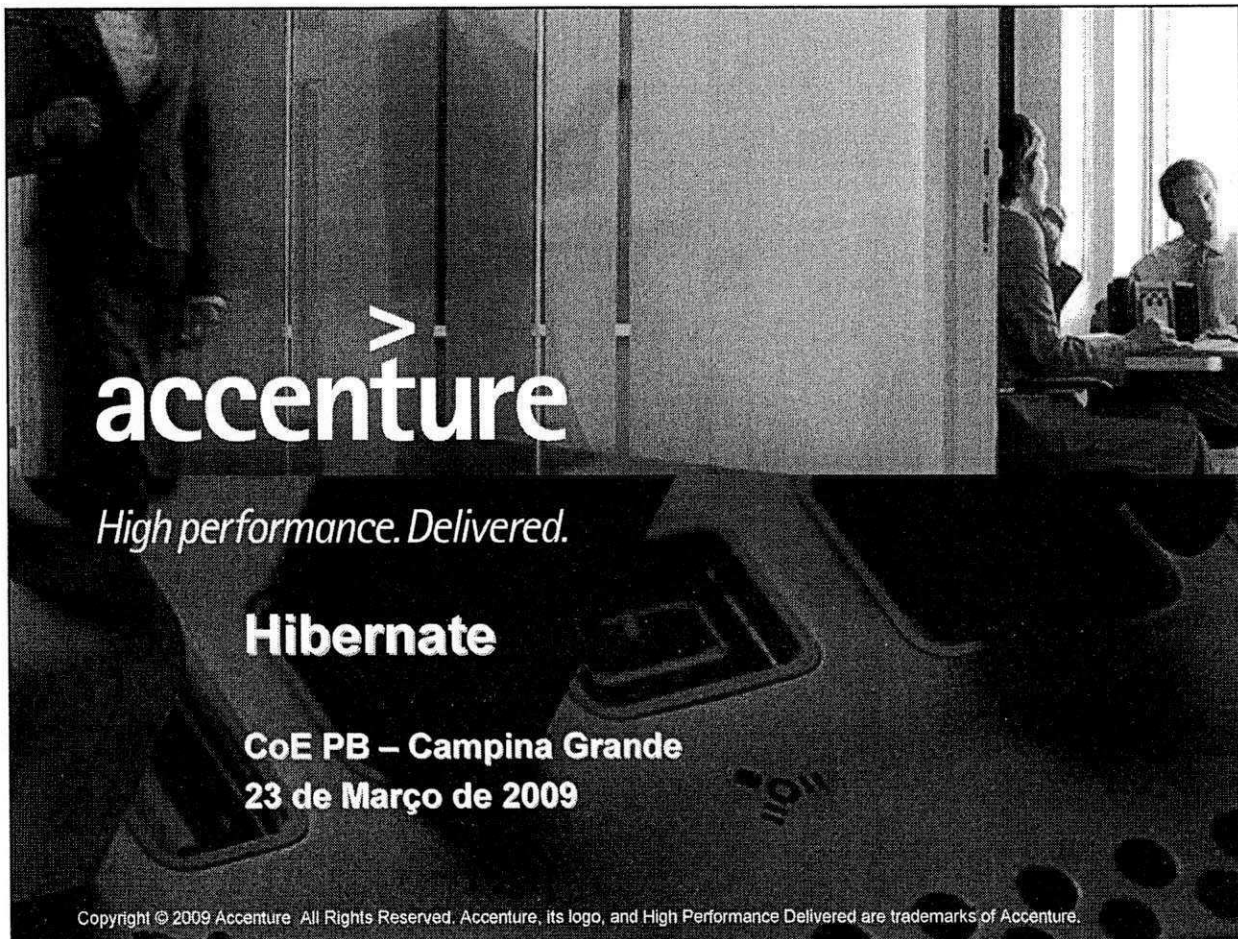
Francisco Manoel de Barros Neto
Supervisor Técnico

Joseluze de Farias Cunha
Supervisora Acadêmica

Joseana Macêdo Fchine
Coordenadora da Disciplina Estágio Integrado

Apêndice B

Hibernate



**>
accenture**

High performance. Delivered.

Hibernate

CoE PB – Campina Grande
23 de Março de 2009

Copyright © 2009 Accenture All Rights Reserved. Accenture, its logo, and High Performance Delivered are trademarks of Accenture.



O que veremos?

- Nesta apresentação você vai conhecer um pouco sobre os mapeamentos do Hibernate em xml. Assumimos que você já tenha um conhecimento básico sobre a linguagem Java, UML, bancos de dados e XML, além de ser capaz de montar um ambiente de desenvolvimento Java com todas as configurações necessárias.



O que é Hibernate?

- O Hibernate é uma ferramenta de mapeamento objeto/relacional para Java. Ela transforma os dados tabulares de um banco de dados em um grafo de objetos definido pelo desenvolvedor.



Vantagens

- Abstração do código SQL, permitindo escolher o tipo de banco de dados durante a execução do programa e mudar a base sem alterar nada no código Java.
- Criação de tabelas do banco de dados de um jeito bem simples.
- Caso o plano de ação padrão tomado pelo Hibernate não satisfaça as necessidades da empresa (como o uso de select *, joins etc), ele possui dezenas de otimizações que podem ser feitas para atingir tal objetivo.

Configurando o Ambiente

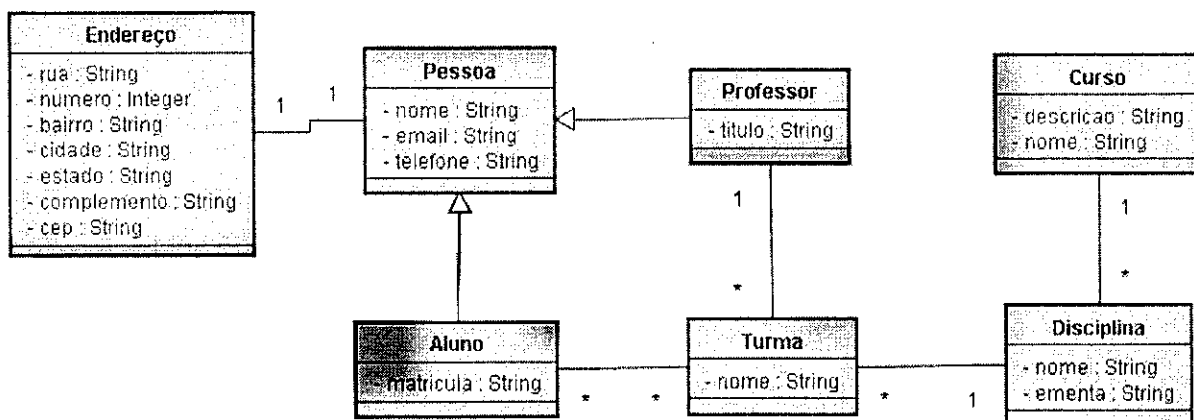
- A versão utilizada nesta apresentação foi a 3.0, você pode baixa-la em: [Hibernate](#)
- Adicione o arquivo hibernate3.jar e os seguintes arquivos da pasta "lib" do download ao seu classpath:
- ehcache-1.1.jar
- jta.jar
- xml-apis.jar
- commons-logging-1.0.4.jar
- c3p0-0.8.5.2.jar
- asm-attrs.jar
- log4j-1.2.9.jar
- dom4j-1.6.jar
- antlr-2.7.5H3.jar
- cglib-2.1.jar
- asm.jar
- jdbc2_0-stdext.jar
- xerces-2.6.2.jar
- commons-collections-2.1.1.jar

Configurando o Ambiente

- O banco de dados escolhido para esta apresentação foi o MySQL x.x, mas os scripts SQL para gerar as tabelas podem ser facilmente adaptados para outros bancos.
- Para usar o MySQL você também vai ter que colocar o driver JDBC dele no seu classpath, você pode baixá-lo em: [Driver JDBC](#)
- Além do driver do MySQL, você também vai ter que adicionar os arquivos .JAR que estão na pasta "lib" do arquivo do driver, o "aspectjrt.jar" e o "aspectjtools.jar".

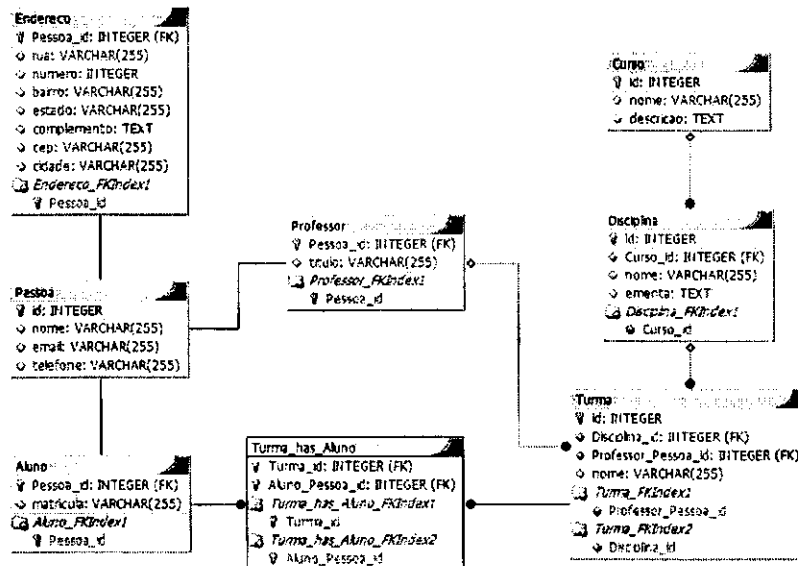
Definindo os objetos do modelo e as tabelas do banco

- Abaixo o modelo que será seguido:



Definindo os objetos do modelo e as tabelas do banco

- Com um modelo de objetos criado, vejamos como essas classes poderiam ser transformadas em tabelas, neste diagrama:



Mapeando as classes para as tabelas



- Classe Pessoa:

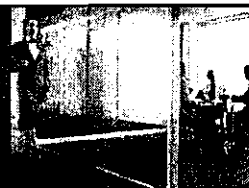
```
public class Pessoa {  
    private String nome;  
    private String email;  
    private String telefone;  
    private Endereco endereco;  
    private Integer id;  
  
    //métodos getters e setters das propriedades  
}
```

- Dica: Para propriedade que vai representar a chave, neste caso, ID, sempre use o tipo Integer, pois trabalhando com objetos, o mapeamento e a resolução se um objeto existe ou não no banco de dados torna-se mais simples para o Hibernate.

Mapeando as classes para as tabelas

- Use seu conhecimento de Java e crie as outras classes de baseando-se no slide 7, lembrando de acrescentar o atributo ID, do tipo INTEGER em todas as classes.
- **Dica:** Quando estiver mapeando as suas classes do modelo para o banco de dados, tente usar os mesmos nomes das classes e de suas propriedades, isso vai evitar várias dores de cabeça. Além do que, se os nomes forem iguais, você não precisa repetir os nomes das tabelas e dos relacionamentos no mapeamento do Hibernate.

Conhecendo o arquivo "hmb.xml" e mapeando um relacionamento 1:1 - Pessoa



```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="Pessoa">

        <!-- Identificador da classe -->
        <id name="id">
            <generator class="increment" />
        </id>

        <!-- Propriedades da classe -->
        <property name="nome" />
        <property name="telefone" />
        <property name="email" />

        <!-- Relacionamento da classe -->
        <one-to-one name="endereco" class="Endereco"
            cascade="save-update" />
    </class>
</hibernate-mapping>
```

Conhecendo o arquivo "hmb.xml" e mapeando um relacionamento 1:1 - Endereco

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping
PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="Endereco">
    <id name="id" column="Pessoa_id">
      <generator class="foreign">
        <param name="property"> Pessoa</param>
      </generator>
    </id>
    <property name="bairro" />
    <property name="cidade" />
    <property name="complemento" />
    <property name="estado" />
    <property name="numero" />
    <property name="rua" />
    <property name="cep" />
    <one-to-one name="Pessoa" class="Pessoa" constrained="true" />
  </class>
</hibernate-mapping>
```

Mapeando herança

- No Hibernate existem diversas formas de se fazer o mapeamento de uma relação de herança.
- Neste tutorial foi usada a forma mais comum e mais simples de ser mantida, uma tabela para cada classe.

Mapeando herança - Professor

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <joined-subclass name="Professor" extends="Pessoa">
        <key column="Pessoa_id" />
        <property name="titulo" />
        <set name="turmas" inverse="true">
            <key column="Pessoa_Professor_id" />
            <one-to-many class="Turma" />
        </set>
    </joined-subclass>
</hibernate-mapping>
```

Mapeando herança - Aluno

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <joined-subclass name="Aluno" extends="Pessoa">
        <key column="Pessoa_id"/>
        <property name="matricula"/>
        <set name="turmas" table="Turma_has_Aluno" inverse="true">
            <key column="Aluno_Pessoa_id"/>
            <many-to-many class="Turma" column="Turma_id"/>
        </set>
    </joined-subclass>
</hibernate-mapping>
```

Configurando o Hibernate 3

```
<!DOCTYPE hibernate-configuration PUBLIC
"--//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">
      org.hibernate.dialect.MySQLDialect
    </property>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver
    </property>
    <property name="hibernate.connection.url">
      jdbc:mysql://localhost:3306/hibernate
    </property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">admin</property>
    <!-- Configurações de debug -->
    <property name="show_sql">true</property>
    <property name="hibernate.generate_statistics">true</property>
    <property name="hibernate.use_sql_comments">true</property>
    <mapping resource="Curso.hbm.xml" />
    <mapping resource="Disciplina.hbm.xml" />
    <mapping resource="Turma.hbm.xml" />
    <mapping resource="Pessoa.hbm.xml" />
    <mapping resource="Aluno.hbm.xml" />
    <mapping resource="Professor.hbm.xml" />
    <mapping resource="Endereco.hbm.xml" />
  </session-factory>
</hibernate-configuration>
```

configurando e abrindo sessões do Hibernate

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtility {
    private static SessionFactory factory;
    static {
        try {
            factory = new Configuration().configure().buildSessionFactory();
        } catch (Exception e) {
            e.printStackTrace();
            factory = null;
        }
    }
    public static Session getSession() {
        return factory.openSession();
    }
}
```

Conclusões

- O Hibernate é um framework, que facilita o desenvolvimento de aplicações que acessam bancos de dados, fazendo com que o programador se preocupe mais com o seu modelo de objeto e seus comportamentos, do que com as tabelas do banco de dados.

*observar que
"no texto" é
"dito" Apêndice.*

parte deste c

consta

Anexo C

Processo de Desenvolvimento

C.1 ADM

O CoE utiliza a Metodologia de Desenvolvimento de software da Accenture, o ADM (Accenture Delivery Methods), que é um dos elementos chaves do ADS (Accenture Delivery Suite) da empresa. Os métodos do ADM são consistentes com o Capability Maturity Model (CMM), garantindo a cobertura de todos os Key Process Áreas (KPA) nos níveis II e III do CMM.

Apesar da possibilidade de se retornar a fases anteriores do processo de desenvolvimento, um dos conceitos chaves envolvidos na metodologia é o conceito de contenção de etapa. Todos os testes, validações e verificações são efetuados de maneira a reduzir ao máximo a possibilidade de que um problema gerado em uma etapa do processo passe para a etapa seguinte.

C.2 CMMi

O Modelo Integrado de Maturidade da Capabilidade de Software (CMMI) fornece às organizações de software um guia de como obter controle em seus processos para desenvolver e manter software e como evoluir em direção a uma cultura de engenharia de software e excelência de gestão.

A melhoria contínua de processo é baseada em muitas etapas evolutivas pequenas ao invés de fundamentar-se em inovações revolucionárias. O CMMI fornece uma estrutura para organizar essas etapas evolutivas em cinco níveis de

maturidade que estabelecem fundamentos sucessivos para a contínua melhoria do processo.

Esses cinco níveis de maturidade definem uma escala ordinal para medir a maturidade de um processo de software da organização e para avaliar a sua capacidade de processo de software. Os níveis também ajudam uma organização a priorizar seus esforços de melhorias.

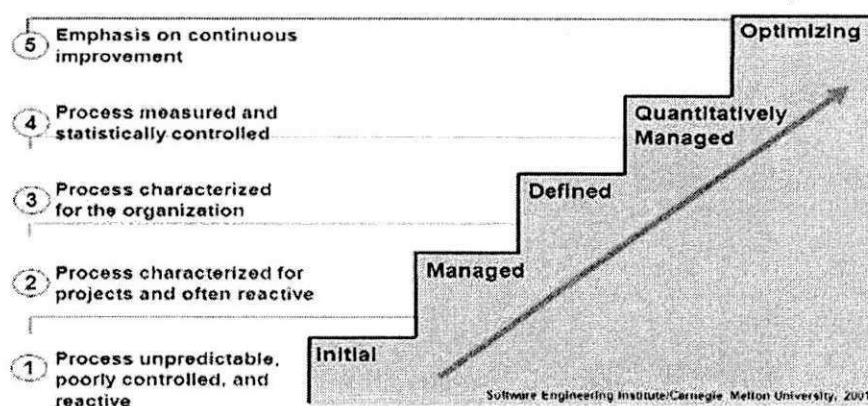


Figura C.1: Níveis de Maturidade

C.3 V-Model

O projeto CoE usa o V-Model para executar atividades de verificação e validação. O V-Model é um framework usado para estabelecer estágios bem definidos para organizar a verificação e a validação por todas as fases do desenvolvimento.

O desenvolvimento começa no lado esquerdo do V-Model com as atividades da análise e de projeto. A especificação é realizada de cima para baixo, tomando decisões e adicionando mais detalhe em cada estágio novo da especificação. Quando o projeto está completo, os processos da construção começam. Uma vez que a construção está completa, o produto move-se através das atividades de verificação, validação e teste no lado de mão direita do V. Durante os estágios iniciais dos testes, o foco está em componentes individuais. À medida que o teste progride, o foco passa para a funcionalidade e a realização do negócio.

O processo de verificação e validação é uma tentativa de tratar os problemas o mais cedo possível no ciclo de vida do desenvolvimento e de assegurar que as especificações estejam completas, corretas e adequadas aos padrões. O processo de testar assegura que as especificações tenham sido corretamente implementadas e que a solução atenda as exigências do negócio e de desempenho.

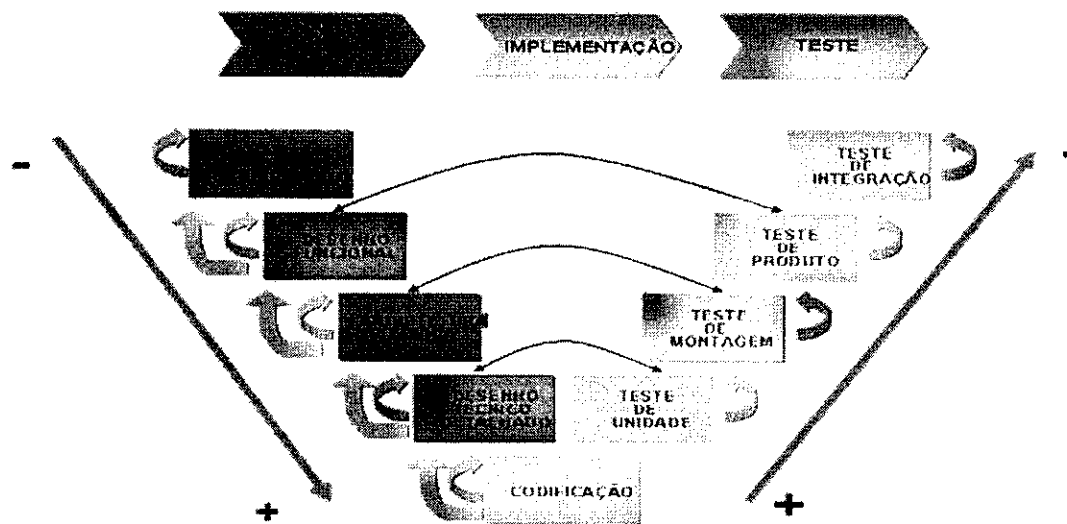


Figura C.2: Framework V-Model

C.4 Metodologia Desenvolvimento CoE

No CoE a fase de análise de requerimento é realizada no cliente pela própria equipe do cliente.

Após receber o documento com a análise de requerimento, a equipe do CoE responsável pela área do requerimento elabora a Análise de Impacto.

O cliente aprovando o desenvolvimento inicia-se a fase do desenho funcional (FD).

Ao término do FD, pode-se paralelizar a elaboração do desenho técnico (TD) e dos scripts de testes de produtos (PT).

Ao término do TD, pode-se paralelizar a elaboração dos scripts de Teste de Montagem (AT), a codificação, testes unitários e a elaboração do Manual de Operação e Change Form.

O teste de integração é especificado, caso necessário, junto ao documento referente ao teste de produto. Normalmente, o teste de integração é realizado no cliente pela própria equipe do cliente ou pela equipe da Accenture, quando é possível.

O CoE possui uma documentação padrão para cada fase do V-Model.

Todo artefato deve ser revisado, seguindo o procedimento de Peer Review.

Para realizar a revisão, é necessário seguir um checklist de Peer Review, que contem todas as informações que dever ser avaliadas no documento.

C.5 Benefícios

Ao usar o V-Model durante todo o projeto, pode-se obter principalmente os seguintes benefícios:

- Melhoria na qualidade e confiança
- Redução da quantidade de re-trabalho;
- Redução do custo de correção do problema;
- Entrega no tempo acordado.

C.6 QPI

O QPI é um programa que visa garantir que todos os projetos seguem o processo do ADM com nível mínimo de maturidade 3 do CMMi. A execução das práticas do QPI são essenciais para um projeto bem sucedido. Estas práticas aplicam-se tanto quando usa-se o Accenture Delivery Methods quanto para os métodos do cliente.

Periodicamente é realizado uma auditoria no projeto, fazendo parte do processo a coleta de evidências e entrevistas com membros do CoE.

A equipe de Qualidade é responsável por preparar o projeto para a auditoria, ou seja, mantém os processos sempre vivos e dinâmicos, além dos planos atualizados. Garante que os membros do projeto têm conhecimento do processo como um todo através de auditorias internas.

Os planos seguidos pelo QPI e que devem ser do conhecimento de todos são (disponíveis no Portal da Qualidade):

C.7 Business Continuity Approach

É o Plano de Contingência, onde descrevemos o que deve ser feito em caso de uma tragédia/desastre, ou seja, em caso de acidentes no projeto e/ou na Accenture que cause a parada no desenvolvimento para o cliente.

C.8 Communication Plan

No Plano de Comunicação descrevemos todos os tipos de comunicação realizadas no projeto. Exemplos: emails enviados, PortalCoE, reuniões etc.

C.9 Configuration Management Plan

A Gerência de Configuração (CM) permite o gerenciamento controlado e repetível dos componentes da Tecnologia de Informação (IT), pois envolve todas as fases do desenvolvimento e manutenção. CM implementa um processo no qual os integrantes do projeto e todos os envolvidos identificam, comunicam, implementam, documentam e gerenciam as mudanças. Quando devidamente implementado, CM mantém a integridade de todos os itens que forem sendo colocados sob seu controle.

C.10 Knowledge Management Plan

O Plano de Gestão do Conhecimento tem o propósito de estabelecer como compartilhar, armazenar, gerenciar, organizar os conhecimentos do Projeto CoE. Além disso, determinar qual conteúdo deve ser disponibilizado para uso dos recursos humanos do projeto.

C.11 Quality Management Plan

O gerenciamento da qualidade é a abordagem sistemática que consiste em processos, procedimentos, técnicas e padrões para garantir:

- A qualidade dos produtos gerados pelo projeto;
- A aplicação de melhores práticas e técnicas que minimizam os riscos com prazos, budget e qualidade;
- O entendimento das expectativas dos stakeholders, com a clara definição da abordagem para atendimento dessas expectativas.

C.12 Risk Management Plan

O objetivo do Plano de Gerenciamento de Riscos é orientar a necessidade de gerenciar e monitorar riscos para evitar atrasos, aumento de custo, problemas de desempenho, impactos ambientais adversos ou outras consequências indesejáveis.

C.13 Unit Operating Model

A leitura do Unit Operating Model do projeto CoE é obrigatória para todos os seus membros. O Unit Operating Model é o guia geral da unidade. Ele descreve o overview da unidade, os processos padrões, e a estrutura organizacional. O Unit Operating Model deve ser visualizado por todos os membros e deve ser uma fonte contínua de informações para todos os membros. Ele fornece um overview dos processos da unidade como também informações específicas sobre outros documentos.

C.14 Unit Measurement Plan

O propósito do Plano de Métricas do Projeto CoE é definir as métricas que irão dar suporte à gerência para tomar decisões baseadas em informações para promover qualidade, produtividade e melhoria de processo. Este plano assegura que as métricas definidas são alinhadas aos objetivos do negócio e do programa, e que as métricas são implantadas em uma abordagem organizada e planejada.