

SILAS DIAS ROMANHA

**UM MODELO DE FÁBRICA DE SOFTWARE EM INSTITUIÇÕES DE
ENSINO SUPERIOR**

Guaratinguetá - SP
2016

Silas Dias Romanha

**UM MODELO DE FÁBRICA DE SOFTWARE EM INSTITUIÇÕES DE ENSINO
SUPERIOR**

Dissertação de Mestrado apresentado ao Conselho do Curso de Mestrado Profissional em Engenharia de Produção da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Mestrado Profissional em Engenharia de Produção.

Orientador: Prof. Dr. José Roberto Dale Luche
Co-orientador: Prof. Dr. Jorge Muniz Júnior

Guaratinguetá
2016

R758u	<p>Romanha, Silas Dias Um Modelo de Fábrica de Software em Instituições de Ensino Superior / Silas Dias Romanha – Guaratinguetá, 2016. 106p. f : il. Bibliografia: f. 59-65</p> <p>Dissertação (Mestrado) – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2016. Orientador: Prof. Dr. José Roberto Dale Luche Coorientadora: Prof. Dr. Jorge Muniz Junior</p> <p>1. Software – desenvolvimento. 2. Ensino superior. I. Título</p> <p style="text-align: right;">CDU 681.3.06(043)</p>
-------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Silas Dias Romanha

**ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE
“MESTRE EM ENGENHARIA DE PRODUÇÃO”**

**PROGRAMA: ENGENHARIA DE PRODUÇÃO
CURSO: MESTRADO PROFISSIONAL**

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO


Prof. Dr. Jorge Muniz Junior
Coordenador

BANCA EXAMINADORA:


Prof. Dr. JOSÉ ROBERTO DALE LUCHE
Orientador / UNESP-FEG


Prof. Dr. ANDREIA MARIA PEDRO SALGADO
UNESP-FEG


Prof. Dr. CARLOS JOSÉ PEREIRA
IFSP/São Carlos

Mai de 2016

DADOS CURRICULARES

Silas Dias Romanha

NASCIMENTO	20.12.1987 – BELO HORIZONTE / MG
FILIAÇÃO	Wagner de Souza Romanha Telma Dias Romanha
2006/2009	Curso de Graduação Sistemas de Informação – Associação Educacional Dom Bosco
2010/2012	Curso de Pós-Graduação em Gestão de Negócios e Finanças, na Associação Educacional Dom Bosco.

de modo especial, à minha esposa Alanna, sem a qual eu não teria sido capaz de chegar onde cheguei e que sempre me apoiou em todos os momentos.

“Cada sonho que você deixa para trás, é um pedaço do seu futuro que deixa de existir.”

Steve Jobs

ROMANHA, S. D. UM MODELO DE FÁBRICA DE SOFTWARE EM INSTITUIÇÕES DE ENSINO SUPERIOR. 2016. Dissertação de Mestrado Profissional (Mestrado Profissional em Engenharia de Produção) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2016.

RESUMO

Este trabalho aborda os aspectos relacionados à implantação de Fábrica de Software (FS) em Instituições de Ensino Superior (IES) no Brasil e busca identificar fatores que influenciam tais instituições na decisão de adotar o referido modelo, assim como os fatores de risco e dificuldades normalmente encontradas durante o processo. A análise realizada neste estudo permite que IES possam se atentar a aspectos que facilitem a implementação de uma Fábrica de Software em seu ambiente acadêmico. O trabalho utiliza como método a pesquisa de campo, a pesquisa documental e apresenta uma análise dos resultados com as instituições pesquisadas. A partir dos resultados observados nestas IES, é apresentada uma proposta de implementação e gerenciamento de Fábrica de Software Acadêmica (FSA), aprovada e implementada na Associação Educacional Dom Bosco (AEDB), incluindo seus resultados parciais.

PALAVRAS-CHAVE: Fábrica de Software. Métodos ágeis de Desenvolvimento de Software. Scrum. Instituição de Ensino Superior.

ROMANHA, S. D. A MODLE OF SOFTWARE FACTORY IN HIGHER EDUCATION INSTITUTION. 2015. Master's Degree Dissertation (Master's Degree in Industrial Engineering) - Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2015.

ABSTRACT

This paper addresses the issues related to Software Factory deployment (FS) in Higher Education Institutions (HEIs) in Brazil and aims to identify factors that influence such institutions in the decision to adopt that model, as well as the risk factors and difficulties usually during the process. The analysis in this study allows IES can pay attention to aspects that facilitate the implementation of a Software Factory in their academic environment. The work uses as a method of field research, desk research and analyzes the results of the surveyed institutions. The results observed in these HEIs, the study presents a proposal for deployment and management Academic Software Factory (FSA), approved and implemented in Associação Educacional Dom Bosco (AEDB), including its partial results.

KEYWORDS: Software factory. Agile Software Development. Scrum. Higher Education Institution.

LISTA DE FIGURAS

Figura 1 – Justificativa de negócio para o projeto resultante da pesquisa.....	18
Figura 2 – Escopo de Fábrica de Software	23
Figura 3 – Etapas da Pesquisa Realizada.....	32
Figura 4 – Membros das equipes de projeto FSA da AEDB.....	48
Figura 5 – Ciclo máximo de permanência dos alunos na Fábrica de Software.....	49
Figura 6 – Origens de projetos da FSA da AEDB.....	50
Figura 7 – Cronograma macro da FSA da AEDB	53
Figura 8 – Tela principal do primeiro software desenvolvido pela FSA da AEDB	55
Figura 9 – Tela principal do segundo software desenvolvido pela FSA da AEDB	56

LISTA DE GRÁFICOS

Gráfico 1 – Metodologias ágeis mais utilizadas atualmente no mundo	37
Gráfico 2 – Motivação de IES ao implementar FSA.....	40
Gráfico 3 – Principais dificuldades relatadas ao implementar uma FSA	43

LISTA DE TABELAS

Tabela 1 – Exemplos de Fábricas de Software no Brasil	16
Tabela 2 – Evolução do Conceito de Fábrica de Software.....	22
Tabela 3 – Exemplos de Fábricas de Software em IES no Brasil	38
Tabela 4 – Fatores que influenciam para a implementação de Fábrica de Software em IES..	41
Tabela 5 – Principais dificuldades encontradas por cada IES durante o processo de implementação de Fábrica de Software.....	44
Tabela 6 – Vantagens da participação na FSA da AEDB por perfil de aluno.....	47

LISTA DE ABREVIATURAS E SIGLAS

AEDB	Associação Educacional Dom Bosco
AMP	Avaliação e Melhoria do Processo Organizacional
APG	Adaptação do Processo para Gerência de Projeto
AQU	Aquisição
ARQ	Análise e Resolução de Causas
CenPRA	Centro de Pesquisas Renato Archer
CESAR	Centro de Estudos e Sistemas Avançados de Recife
CMM	Capability Maturity Model
CMMI	Capability Maturity Model - Integration
DEP	Desempenho do Processo Organizacional
DFP	Definição de Processo Organizacional
DRE	Desenvolvimento de Requisitos
FS	Fábrica de Software
FSA	Fábrica de Software Acadêmica
GCO	Gerência de Configuração
GPR	Gerência de Projetos
GQA	Gerência da Qualidade
GQP	Gerência Quantitativa do Projeto
GRE	Gerência de Requisitos
GRI	Gerência de Riscos
IES	Instituição de Ensino Superior
IIO	Inovação e Implantação na Organização
INEP	Instituto Nacional de Estudos e Pesquisas
ISO	International Organization for Standardization
ISP	Instalação do Produto
ITP	Integração do Produto
LIP	Liberação do Produto
MCTI	Ministério da Ciência, Tecnologia e Inovação
MEC	Ministério da Educação
MED	Medição
MPS.BR	Melhoria de Processos do Software Brasileiro
MR-MPS	Modelo de referência para melhoria do processo de software
PIB	Produto Interno Bruto
POO	Programação Orientada à Objetos
RUP	Rational Unified Process
SEI	Software Engineering Institute
SGDB	Sistema Gerenciador de Banco de Dados
SPYCE	Software Process Improvement and Capability Determination
STE	Solução Técnica
SW	Software
TI	Tecnologia da Informação
TRE	Treinamento
VAL	Validação
VER	Verificação
XP	Extreme Programming

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS E DELIMITAÇÃO.....	16
1.2	JUSTIFICATIVAS	17
1.3	ORGANIZAÇÃO DO TRABALHO	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	HISTÓRICO DO DESENVOLVIMENTO DE SOFTWARE.....	20
2.2	FÁBRICA DE SOFTWARE	21
2.2.1	Recursos Humanos em Fábricas de Software	25
2.2.2	As Fábricas de Software no mundo	28
2.2.3	Perspectiva para as Fábricas de Software no Brasil	29
2.2.4	Fábricas de Software Acadêmicas	29
3	MÉTODO DE PESQUISA	31
3.1	A ASSOCIAÇÃO EDUCACIONAL DOM BOSCO	31
3.2	MÉTODO UTILIZADO.....	32
3.2.1	Definição do projeto	33
3.2.2	Pesquisa Documental e Elaboração de Objetivos Específicos	33
3.2.3	Delimitação da Pesquisa	34
3.2.4	Elaboração e Envio de Questionário	34
3.2.5	Elaboração da Proposta	34
3.2.6	Aplicação do Modelo Elaborado e Análise dos Resultados Parciais	35
4	FÁBRICAS DE SOFTWARE EM INSTITUIÇÕES DE ENSINO SUPERIOR NO BRASIL	36
4.1	ASPECTOS BEM SUCEDIDOS DAS FSA PESQUISADAS.....	36
4.2	MOTIVAÇÃO PARA A IMPLEMENTAÇÃO DE UMA FSA	39
4.3	PRINCIPAIS DIFICULDADES ENFRENTADAS PELAS IES DURANTE O PROCESSO DE IMPLEMENTAÇÃO DE SUAS FSA	42
5	PROPOSTA DE FÁBRICA DE SOFTWARE PARA A ASSOCIAÇÃO EDUCACIONAL DOM BOSCO (AEDB)	45
5.1	O CURSO DE SISTEMAS DE INFORMAÇÃO	45
5.2	A PROPOSTA.....	46
5.2.1	Membros da Fábrica de Software da AEDB	46
5.2.2	Ciclo de Permanência dos Alunos n° FSA	49

5.2.3	Metodologias e Procedimentos	50
5.2.4	Impacto Acadêmico do Projeto	52
5.2.5	Cronograma e Planejamento	53
5.3	RESULTADOS PARCIAIS	53
5.3.1	Projeto de Interface Entre os Sistemas Gennera e TopAcesso.....	54
5.3.2	Projeto de Interface Entre os Sistemas Gennera e Moodle	56
6	CONCLUSÃO.....	58
6.1	SUGESTÃO PARA TRABALHOS FUTUROS	58
	REFERÊNCIAS	59
	APÊNDICE A - QUESTIONÁRIO APLICADO A GESTORES DE FÁBRICAS DE SOFTWARE EM OUTRAS INSTITUIÇÕES DE ENSINO SUPERIOR..	66
	APÊNDICE B - ESTUDOS DE CASO: FÁBRICAS DE SOFTWARE EM INSTITUIÇÕES DE ENSINO SUPERIOR NO BRASIL.....	72
	APÊNDICE C - METODOLOGIAS E BOAS PRÁTICAS DE DESENVOLVIMENTO DE SOFTWARE.....	79
	APÊNDICE D – FORMULÁRIO DE INSCRIÇÃO DE ALUNOS PARA TRABALHO REMOTO VOLUNTÁRIO NA FÁBRICA DE SOFTWARE DA AEDB.....	89
	APÊNDICE E – PESQUISA DE SATISFAÇÃO DOS PRODUTOS GERADOS PELA FÁBRICA DE SOFTWARE DA AEDB.....	92
	APÊNDICE F - TABULAÇÃO DA REVISÃO LITERÁRIA	93
	ANEXO A – EMENTA DA DISCIPLINA “METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS” DO CURSO DE SISTEMAS DE INFORMAÇÃO.....	104

1 INTRODUÇÃO

Sistemas de Informação são componentes importantes em atividades de negócio, uma vez que influenciam na vantagem competitiva organizacional. Na medida em que os mercados crescem, aumenta também a demanda por softwares que os atendam em seus diversos aspectos. Essa demanda crescente faz com que as empresas do ramo de desenvolvimento de software busquem formas mais eficientes de produção, ou seja, com maior produtividade, melhor qualidade e menor custo.

Diferentes metodologias vêm sendo discutidas e avaliadas no intuito de promover melhorias na gestão do processo de desenvolvimento de software e é notável que algumas vertentes apontam para a utilização dos modelos originários da gestão da manufatura, adequando-se às suas particularidades do setor de software e levando em consideração que muitas atividades em ambos os setores possuem grande similaridade (DORIGAN, 2010).

Neste contexto, as Fábricas de Softwares, que são plantas de desenvolvimento de aplicativos que seguem conceitos similares às plantas industriais, como reaproveitamento de componentes e divisão do processo produtivo em etapas bem definidas, têm representado uma solução de fornecimento de produtos com qualidade e preço competitivos, além de contribuir para a capacitação de mão de obra especializada. Tal modelo de desenvolvimento de software vem recebendo destaque no Brasil devido a instalação de diversos parques de alta capacidade de produção no país (FERNANDES e TEIXEIRA, 2004).

Além de empresas especializadas, a crescente demanda por mão de obra e a necessidade de proporcionar experiência prática aos alunos de cursos de tecnologia fez surgir uma nova modalidade de negócio, a Fábrica de Software Acadêmica, cujas unidades de produção fazem uso dos recursos do corpo acadêmico da instituição de ensino, como laboratórios, alunos e professores.

A Tabela 1 apresenta alguns exemplos de Fábricas de Software bem-sucedidas no Brasil, tanto de empresas especializadas quanto de instituições de ensino:

Tabela 1 – Exemplos de Fábricas de Softwares no Brasil

Fábrica de Software	Ano de inauguração	Destaque
IBM (BRASIL, 2016)	2001	Primeira organização no Brasil a obter o nível mais alto de maturidade do CMMI (Apêndice C)
Accenture (DIGITAL, 2016)	2010	Possui unidades em 8 estados brasileiros e seus produtos atendem setores como telecomunicações, energia e finanças
Universidade Federal do Pará (PACHECO, 2008)	2005	7º lugar dentre as 25 instituições de ensino de ciência e tecnologia da América Latina que atuam na área de Engenharia de Software (2008); 2º lugar no Prêmio Dorgival Brandão Junior de Qualidade e Produtividade de Software do MCT pelo projeto do Software Livre WebAPSEE, em 2007
Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul – Campus Porto Alegre (BORGES, CARVALHO e MORAES, 2012)	2010	Esta Fábrica de Software Acadêmica, por meio de suas diversas ações, atendeu em média 60 alunos por ano, contribuindo para o objetivo de proporcionar experiência prática em Engenharia de Software.

Fonte: Elaborado pelo autor

1.1 OBJETIVOS E DELIMITAÇÃO

O objetivo geral deste trabalho é elaborar um plano de implementação e gerenciamento de Fábrica de Software Acadêmica para a IES AEDB, atentando-se a aspectos relevantes indicados por outras IES que implementaram projetos de mesma natureza.

Objetivos específicos:

- Analisar os aspectos bem-sucedidos das Fábricas de Software Acadêmicas existentes;
- Identificar as principais motivações para a implementação de Fábricas de Software em ambiente acadêmico;
- Elencar as principais dificuldades enfrentadas durante o processo de implementação do conceito.

Este estudo analisa casos de implementação de Fábricas de Software em Instituições de Ensino Superior no Brasil, onde equipes sejam compostas principalmente pelo corpo acadêmico das mesmas.

1.2 JUSTIFICATIVAS

O Brasil ocupava a oitava colocação no ranking do Mercado Mundial de Software e Serviços, de acordo com o estudo publicado em 2014 pela ABES (Associação Brasileira das Empresas de Software) e o IDC (International Data Corporation), sendo que o mercado brasileiro representava 3% do mercado mundial e 47,4% da América Latina, com tendência de crescimento desta representatividade, segundo mesmo estudo. Além dos esforços individuais de cada profissional, é necessário que as empresas, órgãos governamentais e as universidades se esforcem para criarem mecanismos que suportem o crescimento contínuo da indústria de software e que a faça atingir patamares ainda maiores.

A literatura analisada sobre Fábricas de Software em IES indica a oportunidade de aprofundar os estudos sobre adaptação de metodologias de desenvolvimento de software ao ambiente acadêmico, como indicam os estudos realizados na Universidade Federal de Pernambuco (SOARES et al., 2007) e na Universidade Federal de São Carlos (LEITE e LUCRÉDIO, 2014). Outra oportunidade é a análise sobre estrutura organizacional das fábricas de software acadêmicas, pois são encontradas diferenças importantes entre as universidades que implementaram esta solução, conforme apresenta a Tabela 3, no Capítulo 4.

Um estudo comparativo entre as fábricas de software acadêmicas nacionais pode indicar aspectos relevantes a serem considerados na implementação de novas fábricas. Durante a pesquisa não foi possível encontrar estudos nesse sentido nas bases de dados de artigos, portanto, a análise realizada permitirá que, através do estudo comparativo realizado, instituições de ensino superior possam atentar a aspectos que facilitem a implementação de fábrica de software em seu ambiente acadêmico.

Este estudo também contribui com orientações para a adequação de instituições de ensino à necessidade de proporcionar experiência prática aos alunos das disciplinas ligadas à Engenharia de Software, determinação contida nas Diretrizes Curriculares de Cursos da Área de Computação e Informática do Ministério de Educação (CEEInf, 1999), que diz que a instituição de ensino deve possibilitar que o aluno adquira experiência na aplicação destes conceitos por meio da prática em laboratórios e estágios. Além disso, autores como Sarinho (2005) e Teixeira e Cukierman (2005) afirmam que os aspectos técnicos da Engenharia de Software precisam ser trabalhados na prática para garantir a assimilação do conteúdo administrado.

A justificativa de negócio para o projeto resultante da pesquisa, segundo a direção da IES a qual ele se destina, pode ser resumida em quatro aspectos principais:

- Proporcionar experiência prática em Engenharia de Software aos alunos de seu curso de Sistemas de Informação, mantendo o corpo discente atualizado em relação às mais modernas técnicas e ferramentas de desenvolvimento e gerenciamento de projetos de software, além de contribuir para o aprendizado em empreendedorismo e o desenvolvimento de habilidades necessárias para a vida profissional do egresso.
- Atender às demandas internas da instituição, que carece de softwares personalizados para diversos tipos de situações, desde metodologias de ensino, passando pela interação extraclasse entre alunos e o corpo docente, até a controladoria, direção e secretarias da faculdade.
- Contribuir para o desenvolvimento das empresas da região onde a instituição está inserida por meio de soluções sistêmicas de baixo custo, gerando conseqüentemente recursos financeiros a serem revertidos em melhorias na própria fábrica e na instituição.
- O projeto de Fábrica de Software da AEDB vai ao encontro também do planejamento estratégico da instituição, que visa diversificar os serviços prestados pela mesma, garantindo uma maior adaptabilidade a novos cenários e às mudanças pelas quais passa o mercado de ensino superior no Brasil e no mundo.

Figura 1 - Justificativa de negócio para o projeto resultante da pesquisa



Fonte: Elaborado pelo autor.

1.3 ORGANIZAÇÃO DO TRABALHO

Este documento está organizado da seguinte maneira: no próximo Capítulo são apresentados os fundamentos teóricos que servem de base para a realização deste estudo. Nele, primeiramente é apresentado o histórico do desenvolvimento de software, em seguida, o conceito de Fábrica de Software, onde são abordadas suas principais características, incluindo aspectos como recursos humanos, gerenciamento, fatores de sucesso e ferramentas de apoio, assim como um breve panorama da realidade do conceito no Brasil e no exterior. No Capítulo 3 são apresentados os aspectos metodológicos utilizados para a pesquisa. No Capítulo 4 é apresentado o resultado da pesquisa realizada, que visou encontrar e documentar casos de sucesso da implementação de Fábricas de Software em instituições de ensino superior no Brasil. Já no Capítulo 5 é apresentada a proposta de FSA encomendada, aprovada e implementada na AEDB, baseada principalmente nas experiências relatadas pela IES pesquisadas, conforme objetivo geral estabelecido. No Capítulo 6 encontram-se as conclusões e perspectivas de pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo, são apresentados os conceitos abordados no trabalho, começando com a evolução do processo de desenvolvimento de software ao longo das últimas décadas, em seguida, são apresentados os conceitos e características de uma Fábrica de Software.

2.1 HISTÓRICO DO DESENVOLVIMENTO DE SOFTWARE

No final da década de 60, a demanda por novos aplicativos era crescente e os problemas que estes resolviam se tornavam cada vez mais complexos. Este cenário somado a ausência de técnicas bem estabelecidas de desenvolvimento software ficou conhecido como a “Crise do Software”. Como resultado, no ano de 1968 foi criada a disciplina de Engenharia de Software, após uma conferência da OTAN em Garmisch, Alemanha, que tinha como objetivo estabelecer práticas mais maduras para o processo de desenvolvimento. Daí em diante, surgiram diversas técnicas para lidar com equipes de desenvolvedores e com as várias etapas de um projeto de software (REZENDE, 2005).

A partir da década de 90, consolidaram-se três modelos de produção predominantes no desenvolvimento de software, adotados de acordo com o tipo de produto oferecido pela empresa ao mercado, são eles (MOTTA e VASCONCELOS, 2006):

- Empresas de informática que desenvolvem softwares inteiramente personalizados, conhecidas pelo termo *Job-shop organizations*, que elaboram projetos únicos e adaptados especificamente às necessidades de cada cliente. Cada projeto pode utilizar componentes, regras e ferramentas diferentes para seu desenvolvimento. Nessas empresas, as equipes de trabalho podem mudar constantemente, dependendo do projeto em que atuam, os profissionais são altamente qualificados e geralmente são especialistas em um tipo de sistema específico.
- Empresas que trabalham com um modelo flexível de projeto e produção, ou *Flexible design and production system*, por sua vez, fabricam softwares semipadronizados. Também conhecidas como Fábricas Flexíveis de Software, nessas organizações os produtos quase prontos são separados em módulos que são mantidos em estoque e depois adaptados, configurados e finalizados de acordo com as necessidades específicas dos clientes no momento da instalação.
- Fábrica Convencional de Projeto e Produção, ou *Conventional factory production and design*, é o modelo onde os produtos são inteiramente padronizados, com

componentes intercambiáveis. Neste modelo, a produção em massa e os altos volumes são obtidos graças à utilização de procedimentos e formas de produção inteiramente padronizados, o que permite economias de escala significativas.

Em 2001, Kent Beck e outros 16 especialistas em desenvolvimento de software elaboraram um documento que ficou conhecido como o Manifesto para o desenvolvimento ágil de software (SOMMERVILLE, 2011). Este documento foi uma “reação” ao movimento que vinha crescendo dos métodos orientados a processo como o CMM e o CMMI que exigem dos programadores uma disciplina de trabalho muito rígida e burocrática. O documento declarava o seguinte:

Estamos descobrindo melhores formas de desenvolvimento de software fazendo e ajudando outros a fazê-lo. Por meio desse trabalho passamos a valorizar:

- Indivíduos e interações ao invés de processos e ferramentas.
- Software funcionando ao invés de documentação abrangente.
- Colaboração do cliente ao invés de negociação de contratos.
- Resposta a modificações em vez de seguir um plano.

A partir deste manifesto surgiram novas metodologias de desenvolvimento de software como o Extreme-Programming, Scrum, entre outros, abordados no Apêndice C.

2.2 FÁBRICA DE SOFTWARE

Existe uma tendência, já percebida pela indústria de Tecnologia da Informação, que aponta para o fato de que muitas metodologias para a criação de software se assemelham cada vez mais com a abordagem da criação de produtos na gestão da manufatura, apesar de alguns autores, como Armour e Miller (2000), argumentarem que software não seria de fato um produto, mas sim uma forma de armazenamento de conhecimento, de modo que seu desenvolvimento não equivale a produzir um produto, mas adquirir conhecimento. Em paralelo a isso, o mercado consumidor de TI está cada vez mais exigente no que diz respeito aos aspectos relacionados à produtividade, ao custo e a qualidade. As empresas do ramo têm procurado se transformar buscando um modelo que possa suprir eficientemente estas necessidades (OLIVEIRA e NETO, 2003).

Nesse contexto, surgiu o termo “Fábrica de Software”, que traz para o ambiente de desenvolvimento de aplicativos conceitos e metodologias análogas ao processo de produção

fábrica tradicional, baseado em componentes com características semelhantes e com a mesma qualidade.

Historicamente, o conceito de Fábrica de Software, que surgiu em sua forma mais simples em meados da década de 60, vem evoluindo com o passar dos anos, como mostra a Tabela 2:

Tabela 2 - Evolução do conceito de Fábrica de Software

Fase	Características	Período
1	Gerência da estrutura e Organização básica Objetivos da manufatura de software são estabelecidos; Foco no produto é determinado; começa a coleta de dados sobre o processo.	meados dos anos 60
2	Padronização e Customização da Tecnologia Objetivos dos sistemas de controle são estabelecidos; Métodos padrões são estabelecidos para o desenvolvimento; Desenvolvimento em ambiente online; Padronização das habilidades por meio de treinamento de empregados; Bibliotecas de código-fonte são introduzidas; surgem metodologias integradas e ferramentas de desenvolvimento.	início dos anos 70
3	Mecanização e Suporte ao processo Introdução de ferramentas para apoio ao controle de projetos; Introdução de ferramentas para a geração de código, teste e documentação; Integração com ferramentas de banco de dados.	final dos anos 70
4	Refinamento do Processo e Extensão Revisão dos padrões; Introdução de novos métodos e ferramentas; Estabelecimento de controle de qualidade e círculos da qualidade; Transferência de métodos e ferramentas para subsidiárias e terceiros.	década de 80
5	Automação Flexível Introdução de ferramentas de automação de design; Introdução de ferramentas de apoio à reutilização; Introdução de ferramentas de apoio à análise de requisitos; Integração de ferramentas em plataformas de desenvolvimento; Disseminação de metodologias ágeis de desenvolvimento, como Scrum.	a partir da década de 90

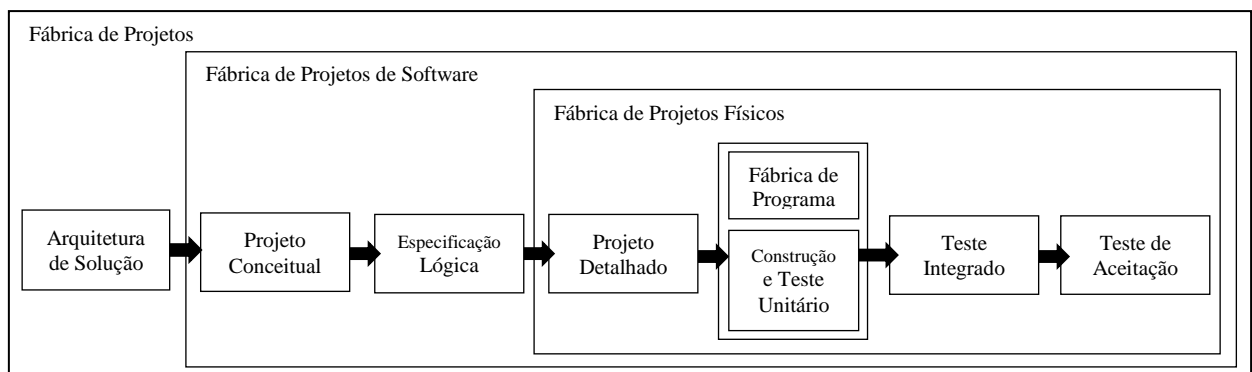
Fonte: (FERNANDES e TEIXEIRA, 2004)

Assim como nas organizações fabris idealizadas por Taylor, Fayol e Ford no século XX, nas fábricas de software também há uma busca pela qualidade, produtividade e baixo custo de produção, naturalmente fazendo com que vários paradigmas encontrados no processo fabril tradicional também estejam presentes no dia a dia das operações de uma fábrica de software.

De maneira análoga ao processo de fabricação de produtos em linhas de montagem, os padrões atuais de desenvolvimento de software, como a Programação Orientada a objetos (POO), determinam que os softwares sejam compostos por módulos ou partes menores que são acopladas para a montagem do produto final, possibilitando que partes individuais possam ser desenvolvidas independentemente das demais. Dessa forma, os projetos de software podem ser planejados em termos estruturais, permitindo a participação de diversos programadores, na própria empresa ou de forma terceirizada, onde cada um recebe requisições de novos processos e produzem componentes com características determinadas e específicas, atendendo a níveis de qualidade previamente determinados, podendo estes módulos ou partes serem posteriormente utilizados em outras soluções, o que é conhecido como reaproveitamento de código (ROCHA, OLIVEIRA e VASCONCELOS, 2004).

Alguns requisitos devem ser previstos durante o planejamento de uma fábrica de software, tais como: definir os perfis funcionais e as respectivas atividades a serem desempenhadas por cada perfil; definir a metodologia de desenvolvimento de software a ser utilizada; definir um plano de processos que inclua a descrição das atividades, relacionando-as com seus respectivos artefatos e perfis funcionais e definir ferramentas de apoio a serem utilizadas. Além disso, a Fábrica de Software pode ter vários escopos, que podem variar desde um projeto completo de software, até mesmo um projeto físico ou somente a codificação de programas, conforme demonstrado abaixo:

Figura 2 - Escopo de Fábrica de Software



Fonte: (FERNANDES e TEIXEIRA 2004).

O sucesso de uma Fábrica de Software passa pela adoção de processos de desenvolvimento que auxiliem na definição e na distribuição das tarefas, na definição dos responsáveis pelas várias etapas do ciclo de vida do software, em uma boa comunicação tanto com o cliente quanto e entre os membros da equipe, na previsão da demanda, em fazer uso de mecanismos de controle e melhoria contínua dos processos, na gestão do conhecimento e de recursos humanos e na utilização de bibliotecas para reutilização de código e componentes. (ROCHA, OLIVEIRA e VASCONCELOS, 2004).

A produtividade é outro fator que determina o sucesso de uma Fábrica de Software. Para Toledo (2008), a produtividade na produção de software pode ser influenciada por três grupos de fatores:

- Fatores tecnológicos, como linguagens de programação, ferramentas de projeto, ambientes de desenvolvimento e capacidade dos equipamentos utilizados.
- Fatores humanos onde são considerados o perfil, formação, motivação, comprometimento e capacitação das pessoas, além do modelo de gerenciamento utilizado.
- Fatores organizacionais referentes aos processos de trabalho utilizados como metodologia, prática gerencial, ambiente físico referente a acomodações, conforto e bem-estar dos recursos humanos.

Ainda segundo Toledo (2008), o uso de ferramentas de apoio é importante para o desenvolvimento das tarefas pertinentes a uma Fábrica de Software. Tais ferramentas visam garantir uma maior produtividade e suporte à comunicação com o cliente. Trata-se de ferramentas como as seguintes:

- Ferramentas de desenvolvimento e modelagem;
- Ferramentas para relatar erros durante o desenvolvimento;
- Ferramentas de gerenciamento de projetos;
- Ferramentas para comunicação instantânea entre os participantes;
- Ferramentas para controle de versão, possibilitando uma melhor organização das atividades e dos documentos do projeto;
- Sistema Gerenciador de Banco de Dados (SGDB).

Para as Fábricas de Software, assim como para as fábricas tradicionais, existem certificações e boas práticas que regem os processos e legitimam a qualidade dos produtos e serviços oferecidos, como o MPS.Br e o CMM/CMMI, abordados em detalhes no Apêndice C.

2.2.1 Recursos Humanos em Fábricas de Software

Na produção industrial, os equipamentos interferem diretamente nas medidas de produtividade, o que requer estratégias que os faça produzir com o mínimo possível de interrupções. Por outro lado, em uma “linha de produção” de softwares, o elemento fundamental é o ser humano. Os recursos humanos em uma fábrica de software, assim como em qualquer outra área, precisam ser planejados e bem treinados para as tarefas que irão desenvolver e alinhados ao tipo de demanda, levando em consideração a natureza e complexidade do trabalho a ser realizado (PMBOK, 2004).

A partir da década de 90, o profissional de sistemas passou por uma mudança de perfil, deixando de ser um especialista e passando a atuar como um generalista, de maneira que hoje em dia, muitas empresas chegam a não utilizar uma definição de cargo detalhada, podendo o profissional assumir diferentes funções, em diferentes projetos, até mesmo simultaneamente. Dessa maneira, o cargo acaba funcionando como mera referência para a remuneração. Para entender melhor esse contexto, é preciso deixar claro o significado de alguns termos (PESSOA FILHO, 2008):

- **Habilidades Técnicas** – Conhecimento prático e teórico adquirido ao longo da formação acadêmica, e em cursos complementares. Inclui metodologias, técnicas, ferramentas metodológicas, linguagens de programação e etc.
- **Habilidades de Negócio** - Estas são adquiridas ao longo do exercício profissional, por meio do desenvolvimento de soluções efetivas para empresas. Podem ser funções empresariais, de administração, processos, procedimentos, idiomas e etc.
- **Habilidades Comportamentais ou Humanas** - Ligadas à educação, cultura, filosofia de vida e com os relacionamentos humanos e corporativos. É possível listar como exemplo destas habilidades, a pró-atividade; a criatividade; a comunicação; a capacidade de expressão e o relacionamento pessoal; o espírito de equipe; a administração participativa; o planejamento pessoal; a capacidade de organização, concentração, atenção, disponibilidade e responsabilidade.

Ainda segundo Pessoa Filho (2008), além de habilidades técnicas e comportamentais, é desejável que os funcionários em uma fábrica de software, assim como nas demais modalidades de empresas que prestam serviços de software, possuam também conhecimentos em outras áreas de negócio, visto que para resolver problemas é preciso primeiramente entendê-los.

Dentre os principais papéis existentes em uma equipe funcional requerida para uma fábrica de software e suas respectivas atividades, é possível destacar (MEDEIROS, ANDRADE, ALMEIDA, ALCUQUERQUE e MEIRA, 2005):

- Gerente de Negócios: responsável pela prospecção do mercado e pela venda dos serviços.
- Analista de Sistemas: a este, cabe realizar o levantamento de requisitos, a análise, definição da arquitetura e a elaboração da documentação do sistema a ser desenvolvido.
- Analista de Qualidade: responsável pela revisão dos artefatos gerados, pelo controle de mudanças, a definição e a validação da qualidade do processo utilizado.
- Engenheiro de Software: garante que o sistema seja implementado de acordo com as especificações em sua documentação e seguindo o processo de desenvolvimento definido.
- Analista de Testes: realiza o desenvolvimento, a validação e a execução de testes de software que visam assegurar a qualidade e integridade do software produzido.
- Líder de Equipe: faz a coordenação e a atribuição de tarefas entre os membros da equipe, fornecendo relatórios periódicos ao gerente de projetos sobre o andamento das atividades.

Em cada equipe de uma Fábrica de Software é recomendado que existisse também a figura do Gerente do Projeto, que é um profissional dedicado exclusivamente a este papel, o que o habilita a atuar fortemente na interface com o cliente do projeto em questão. Segundo o PMBOK (2004), os processos de iniciação, planejamento, execução, monitoramento, controle e encerramento, devem ser integrados e aplicados pelo gerente de projeto. Também fazem parte das atribuições deste profissional identificar as necessidades; estabelecer objetivos; balancear demandas de qualidade, escopo, tempo e custo além de adaptar especificações e planos às necessidades das diversas partes interessadas. O gerente ainda deverá atuar de maneira a garantir que o produto final atenda aos requisitos de qualidade previamente estabelecidos, seguindo métodos e padrões de estimativas baseados em históricos; utilizar métricas específicas para estimar tempos padrões de atendimento levando em consideração a tecnologia, o tamanho e o domínio da demanda; possuir e fazer uso de mecanismos de apuração, apropriação e controle de custos e manter o controle sobre os níveis de serviço.

No que diz respeito ao mercado de trabalho, de acordo com os dados divulgados pela Brasscom - Associação Brasileira de Empresas de Tecnologia da Informação e Comunicação, o setor de Tecnologia da Informação (TI) possui alta demanda por profissionais por um conjunto de fatores:

- Cresce historicamente a taxas superiores ao Produto Interno Bruto (PIB) nacional.
- A TI está inserida em todos os setores da economia moderna, ajudando a aumentar a competitividade e a produtividade das empresas.
- A sociedade utiliza tecnologia cada vez mais intensamente no cotidiano.
- Novas tecnologias – como computação em nuvem, redes sociais, dispositivos móveis e segurança da informação – demandam profissionais com novas habilidades.

A pesquisa da Brasscom, denominada de “O mercado de profissionais de TI” (2014), também aponta que o setor emprega atualmente cerca de 1,3 milhão de profissionais no Brasil. A pesquisa revela também que, até 2020, haverá uma demanda de 750 mil novos trabalhadores, levando em consideração a meta de elevar sua participação no PIB do país para 6,5%. Para o jovem que se especializar nessa área, o resultado poderá ser uma carreira sólida e até mesmo com possibilidade de projeção internacional. A pesquisa apontou também que dez cargos da área representam mais de 90% das contratações no setor no país. E este trabalhador qualificado irá atuar, provavelmente, nas áreas de serviços ou comércio, pois 90,91% da movimentação de saldos da área de TI estão concentradas nestas atividades econômicas. Este dado vale para os oito estados pesquisados: Bahia, Distrito Federal, Minas Gerais, Paraná, Pernambuco, Rio de Janeiro, Rio Grande do Sul e São Paulo. Os cargos com maior demanda no país, segundo a pesquisa, são: analista de desenvolvimento de sistemas; analista de suporte computacional; programador de sistema da informação; técnico em manutenção de equipamentos de informática; help-desk; analista de redes e de comunicação de dados; operador de computador; operador de rede de teleprocessamento; analista de sistemas de automação e programador de internet.

Dessas dez profissões com maior demanda dentro da área de tecnologia de informação, cinco tem sua mão de obra formada pelo do curso de Sistemas de informação: analista de desenvolvimento de sistemas, analista de suporte computacional, programador de sistema da informação, analista de redes e de comunicação de dados e programador de internet. Além disso, o curso fornece conhecimentos básicos também para que seu egresso possa atuar nos demais cargos, apesar de não ser o foco do mesmo.

2.2.2 As Fábricas de Software no Mundo

Um dos principais modelos de negócio para uma fábrica de software, principalmente no exterior, é a exportação. Vários países são considerados grandes exportadores de software, e muitos outros tentam alcançar competitividade neste mercado. Carmel (2003) divide estes países em quatro níveis, levando em consideração três aspectos: maturidade da indústria; quantidade de organizações e o total de exportações. Assim, para o país se enquadrar em cada nível, a indústria de desenvolvimento de software precisa ter:

- Nível 1: mais de 15 anos, centenas de empresas e no mínimo 1 bilhão de dólares em exportação de software. Exemplo de países neste nível: Estados Unidos, Canadá, Reino Unido, Alemanha, França, Bélgica, Holanda, Suécia, Finlândia, Japão, Suíça, Austrália, Irlanda, Israel e Índia.

- Nível 2: mais de 10 anos, pelo menos 100 empresas e no mínimo 200 milhões de dólares em exportação de software. Exemplo de países neste nível: Apenas a Rússia e a China.

- Nível 3: mais de 5 anos, algumas dezenas de empresas e no mínimo 25 milhões de dólares exportados. Exemplo de países neste nível: Brasil, Costa Rica, México, Filipinas, Malásia, Sri-Lanka, Coreia, Paquistão, Romênia, Bulgária, Ucrânia, Polônia, República Tcheca, Hungria, Estônia, Letônia, Lituânia, Eslovênia, Chile, Argentina, Tailândia e África do Sul.

- Nível 4: países que estão começando a amadurecer sua indústria de desenvolvimento de software, e possuem alguma exportação. Exemplo de países neste nível: Cuba, El Salvador, Jordânia, Egito, Bangladesh, Vietnã, Indonésia, Iran e algumas outras nações que não possuem dados disponíveis. Um exemplo em especial merece ser citado, o da Índia. Seu sucesso na área de exportação de software a colocou rapidamente no mesmo patamar dos países mais desenvolvidos neste critério.

Carmel (2003) cita oito fatores para o sucesso nesta área, o que ele chama de “Modelo Oval”:

1. Visão e políticas governamentais, incluindo benefício em impostos e fundos de desenvolvimento.
2. Capital humano, incluindo a orientação e tradição do país, a quantidade, a composição, o conhecimento de línguas e a capacidade gerencial.
3. Remuneração.
4. Qualidade de vida.

5. Afinidades entre indivíduos, grupos de trabalho, entre empresas e entre países, por razões geográficas, culturais, linguísticas ou étnicas.
6. Infraestrutura tecnológica.
7. Fontes externas ou internas de capital.
8. Características da indústria, incluindo efeitos de clusters, número de empresas, o tamanho dessas empresas, as associações que organizam estas firmas, as marcas e os padrões que as empresas buscam alcançar.

2.2.3 Perspectiva para as Fábricas de Software no Brasil

Grandes empresas de tecnologia têm distribuído seus centros de produção de software em países onde seja possível reduzir custos e aumentar a competitividade. Por este motivo o Brasil também tem atraído empresas estrangeiras do ramo. O investimento externo proporciona o surgimento de parques tecnológicos, que surgem por meio de associações de empresas com universidades e órgãos governamentais. O crescimento de um ambiente de pesquisa científica e desenvolvimento tecnológico como este é extremamente benéfico para o país. Bastos (2012) constatou que a isenção fiscal foi o principal fator para atrair investimentos de grandes empresas multinacionais de tecnologia para o país, como a americana IBM, que hoje possui umas das mais modernas fábricas de software do país. Desta forma, é importante incluir esta estratégia no planejamento que visa o aumento da competitividade do país, mantendo os investimentos já realizados e atraindo novos.

Em termos gerais, a Indústria Brasileira de TI está posicionada em 7º lugar no ranking mundial, com um investimento de US\$ 60 bilhões em 2014. O estudo aponta que o Brasil é o 1º lugar no ranking de investimentos no setor de TI na América Latina, representando 46% desse mercado que, só em 2014, somou US\$ 128 bilhões. Um fato a se considerar é a ainda baixa representação das exportações no mercado de software nacional, cerca de 2%, ou US\$225 milhões, segundo estudo publicado em 2014 pela ABES (Associação Brasileira das Empresas de Software), o que pode indicar uma boa oportunidade para as Fábricas de Software nacionais.

2.2.4 Fábricas de Software Acadêmicas

Fábrica de Software Acadêmica (FSA) é uma modalidade de FS cujas unidades de produção são compostas principalmente pelo corpo acadêmico de instituições de ensino superior. Um dos principais objetivos de uma FSA é promover a interdisciplinaridade nos

cursos de TI, agregando em um mesmo ambiente as competências dos diversos componentes do corpo acadêmico, fazendo uso de métodos e processos de criação de software que incluem novas tecnologias, arquiteturas e metodologias de desenvolvimento, estimulando assim o corpo docente e discente a se atualizar constantemente e realizarem atividades de transferência de tecnologia, treinamento empresarial e produção científica.

Outro fator motivador para a implantação de uma Fábrica de Software Acadêmica é a necessidade de proporcionar aos alunos experiência prática associada às disciplinas ligadas à Engenharia de Software, determinação contida nas Diretrizes Curriculares de Cursos da Área de Computação e Informática do Ministério de Educação (CEEInf, 1999), que diz que a instituição deve possibilitar que o aluno adquira experiência na aplicação destes conceitos por meio da prática em laboratórios e estágios. Além disso, autores como Sarinho, Teixeira e Cukierman (2005) afirmam que os aspectos técnicos da Engenharia de Software precisam ser trabalhados na prática para garantir a assimilação do conteúdo ministrado. Já Prikladnicki (2009) afirma que as atividades práticas vivenciadas pelos estudantes fornecem uma assimilação complementar às aulas expositivas.

No Brasil, um bom número de instituições de ensino superior vem adotando o modelo de Fábrica de Software. No Capítulo 4 é apresentado um panorama das experiências relacionadas à FS relatadas por algumas IES no Brasil que se destacam neste cenário.

3 MÉTODO DE PESQUISA

Neste Capítulo, são apresentados os objetos de estudo e procedimentos realizados durante o desenvolvimento da pesquisa. Os objetos de estudo são Instituições de Ensino Superior no Brasil que possuem cursos de computação e laboratórios de produção de software onde atuam membros do corpo acadêmico e onde é aplicado o conceito de Fábrica de Software. O estudo tem como objetivo principal gerar, baseado nas informações coletadas pela pesquisa, um projeto de implementação de Fábrica de Software Acadêmica destinado à Associação Educacional Dom Bosco (AEDB). Para ajudar a compreender o contexto ao qual o projeto se aplica, as características da instituição que o encomendou também serão apresentadas neste Capítulo.

3.1 A ASSOCIAÇÃO EDUCACIONAL DOM BOSCO

A AEDB está instalada na cidade de Resende-RJ, na região Sul Fluminense, que é conhecida por seu potencial turístico e pela presença de grandes indústrias, como as siderúrgicas CSN (Companhia Siderúrgica Nacional) e Votorantim, e as automobilísticas MAN Latin América, Peugeot-Citroen, Nissan, Hyundai e Land Rover.

A instituição concentra em seu Campus três faculdades: Faculdade de Filosofia, Ciências e Letras Dom Bosco - FFCLDB; Faculdade de Engenharia de Resende – FER e Faculdade de Ciências Econômicas, Administrativas e da Computação Dom Bosco FCEACDB, que juntas oferecem 18 cursos de graduação, sendo 5 deles tecnológicos de curta duração, totalizando assim cerca de 2.500 alunos no ensino superior. Por meio do CPGE - Centro de Pesquisa, Pós-Graduação e Extensão, a AEDB oferece diversos cursos de pós-graduação, MBA, em convênio com a Fundação Getúlio Vargas – FGV, além de um mestrado profissional em parceria com a Universidade Estadual Paulista – UNESP. A Associação também mantém o Colégio de Aplicação, com classes de Educação Infantil até o Ensino Médio.

Após conhecer o porte e o ambiente no qual a instituição está inserida, na próxima seção será apresentado o método de pesquisa, visando melhor entender os resultados.

3.2 O MÉTODO UTILIZADO

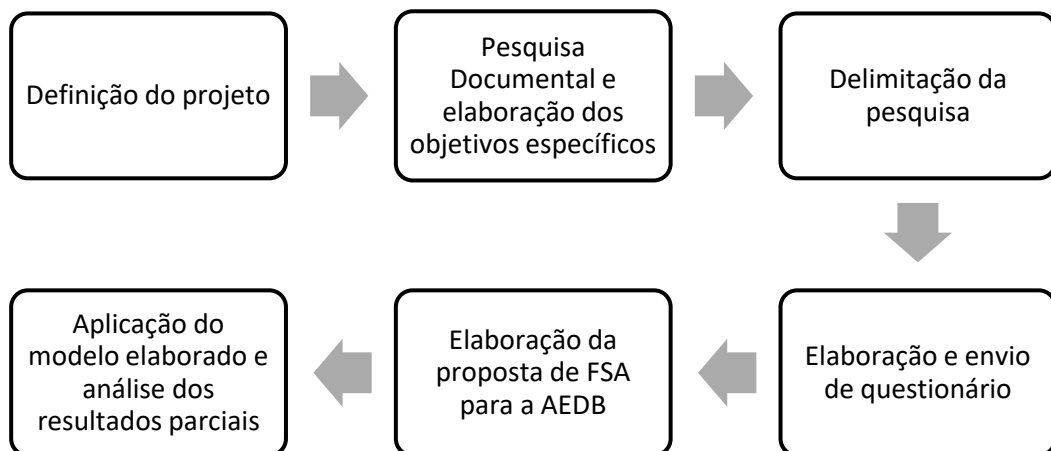
A pesquisa de campo com análise qualitativa se mostrou a melhor opção para o trabalho devido à natureza da investigação. Por meio desse tipo de pesquisa é possível conceituar um problema de forma ampla, ao invés de simplesmente relacionar variáveis quantificáveis. Uma característica importante deste tipo de trabalho é permitir captar aquilo que ainda não foi percebido, assim, são revelados aspectos relevantes que não foram previamente identificados pelo pesquisador (SILVERMAN, 1993; GERHARDT e SILVEIRA, 2009).

O método de pesquisa utilizado é exploratório, pois segundo Quivy e Campenhoudt (1998), é o método mais interessante para os investigadores quando estes estão atuando em uma área nova para eles e possui as seguintes características: os indicadores têm natureza empírica; a partir da observação empírica se desenvolvem os conceitos, novas hipóteses, e depois, o modelo de análise.

Quando o objetivo de uma pesquisa é entender como e por que alguns fenômenos ocorrem, há pouca possibilidade de controle sobre os eventos estudados e o foco de interesse está sobre fenômenos atuais, que só poderão ser analisados dentro de algum contexto da vida cotidiana, é quando o uso de estudos de caso se torna a estratégia de pesquisa mais adequada. Para realizar uma pesquisa deste tipo, é necessário analisar vários exemplos, por isso, o trabalho é um estudo exploratório, abordando um grupo definido de pessoas, dentro de um conjunto de entidades representativo para a área pesquisada (GODOY, 1995).

O fluxograma abaixo mostra as etapas da pesquisa:

Figura 3 – Etapas da pesquisa realizada



Fonte: Elaborado pelo autor.

3.2.1 Definição do Projeto

O projeto surgiu a partir de necessidades relatadas durante entrevistas com a direção da AEDB, que apontou as vantagens que a instituição desejava ao implementar sua própria Fábrica de Software Acadêmica. Após isto, foi realizada uma pesquisa de modo a comprovar a relevância do tema no atual cenário acadêmico. A pesquisa indicou poucos estudos detalhados, principalmente de caráter comparativo entre as diferentes experiências relatadas por outras instituições, por meio de artigos científicos, conforme já explicado no item 1.3.

3.2.2 Pesquisa Documental e Elaboração dos Objetivos Específicos

A maior parte da coleta de dados foi realizada por meio da leitura de artigos científicos, teses, livros e dissertações que descrevem aspectos relevantes relacionados à implementação de Fábricas de Software em instituições de ensino superior no Brasil, assim como artigos que tratam de conceitos importantes para compor a fundamentação teórica necessária para garantir o devido embasamento acadêmico e científico para as propostas.

Também foram consultadas bases de dados e indicadores produzidos por órgãos do governo (federal e estadual), tanto na área de tecnologia quanto na área de educação. Foram utilizados dados do site do MCTI (www.mcti.gov.br), onde se encontram estudos sobre qualidade do software nacional, com pesquisas realizadas nas empresas a cada dois anos, verificando vários aspectos relacionados. A pesquisa também faz uso de dados da ABES - Associação Brasileira das Empresas de Software, que realiza e publica anualmente um estudo com um panorama e tendências do mercado brasileiro de software. Além disso, o estudo conta com dados complementares do MEC, por meio do portal do INEP, para permitir a localização das instituições que atuam com cursos de Sistemas de Informação e potenciais Fábricas de Software, além de outras consultas relacionadas aos cursos de educação técnica e tecnológica.

Após a análise documental, levando em consideração as informações nela obtidas, as necessidades relatadas pela IES AEDB e os relatos documentados por outras IES, foram elaboradas as questões relevantes a serem respondidas pela pesquisa, de maneira a aumentar as chances de sucesso do projeto. Tais questões compõem os objetivos específicos deste estudo, apresentados no Tópico 1.2.

No Apêndice F é apresentado um resumo tabulado dos principais documentos analisados.

3.2.3 Delimitação da Pesquisa

O universo desta pesquisa é composto por instituições de ensino superior no Brasil que tenham documentado experiências relacionadas à implementação de Fábrica de Software no ambiente acadêmico. A amostra foi escolhida pelo critério de tipicidade, que segundo Zanella (2009), melhor se aplica a cenários como o desta pesquisa.

Não foi possível identificar ou mensurar o total de IES no Brasil que se enquadram nos critérios estabelecidos pela pesquisa. Assim, foram analisadas apenas as IES que disponibilizaram em seus sites ou em bases de artigos científicos, material que comprovasse a realização de projetos de implementação ou gerenciamento de Fábricas de Software Acadêmicas.

No Capítulo 4 são apresentadas as IES analisadas, por meio de quadros comparativos.

3.2.4 Elaboração e Envio de Questionário

Devido ao caráter qualitativo do trabalho, também foi elaborado um questionário com perguntas abertas e fechadas, método indicado como uma opção adequada para a coleta de dados, como sugerido por Gerhardt e Silveira (2009). O questionário foi enviado para os principais ocupantes de cargos de chefia ou técnicos diretamente envolvidos com os processos das fábricas de software nas instituições. O questionário utilizado encontra-se no Apêndice A.

Após a análise documental e das respostas ao questionário, foi possível elaborar uma análise de comparação entre as principais características e lições aprendidas pelas Fábricas de Software Acadêmicas identificadas. A análise que serviu de base para a elaboração da proposta apresentada à AEDB pode ser vista no Capítulo 4.

3.2.5 Elaboração da Proposta

A proposta apresentada no Capítulo 5 tem como base, além da fundamentação teórica, a análise elaborada no Capítulo 4, que respondeu às questões de pesquisa levantadas, atendendo assim, aos objetivos específicos deste estudo. Dessa forma, a proposta faz uso das lições aprendidas por outras IES para minimizar as chances de insucesso e melhorar os resultados da implementação de uma Fábrica de Software Acadêmica.

As necessidades e particularidades da AEDB também foram consideradas para a formulação da proposta.

3.2.6 Aplicação da Modelo Elaborado e Análise dos Resultados Parciais

Após elaboração da proposta, a mesma foi apresentada para a direção da instituição e a coordenação do curso de Sistemas de Informação da AEDB, que a aprovaram e colaboraram para o estabelecimento de um cronograma e uma estratégia de implementação do projeto, que pode ser encontrado no Capítulo 5, assim como os resultados parciais alcançados até o momento, conclusões e perspectivas futuras.

4 FÁBRICAS DE SOFTWARE EM INSTITUIÇÕES DE ENSINO SUPERIOR NO BRASIL

Neste capítulo serão apresentadas as IES analisadas, assim como as informações consolidadas a respeito das similaridades e diferenças entre elas no que tange às metodologias adotadas, objetivos esperados, dificuldades encontradas e resultados obtidos. A amostra analisada é composta por dez IES que relataram por meio de artigos científicos suas experiências de implementação e gerenciamento de FSA. As conclusões apresentadas neste Capítulo serviram de base para a elaboração do plano destinado à AEDB, apresentado no Capítulo 5.

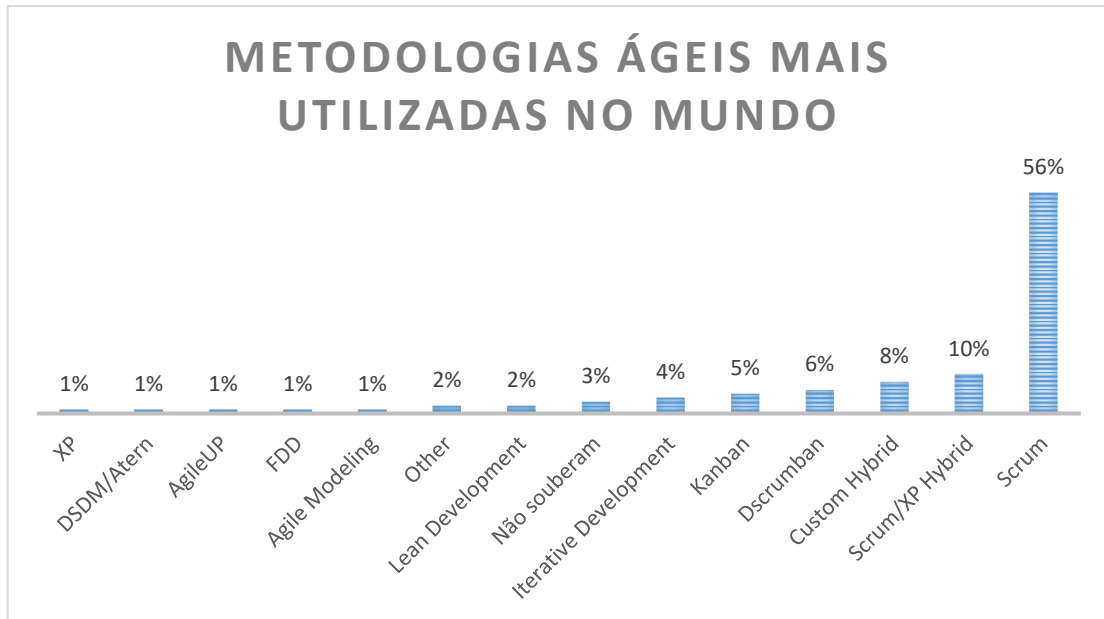
4.1 ASPECTOS BEM SUCEDIDOS DAS FSA PESQUISADAS

De maneira geral, as instituições analisadas optaram por iniciar as atividades de suas FSAs atendendo as demandas internas, pois, segundo elas, dessa maneira é possível aumentar o grau de amadurecimento dos processos estabelecidos enquanto atende a um cliente menos exigente e mais acessível, ou seja, a própria instituição, antes de se comprometer a atender demandas de clientes externos. Tal estratégia contribui também para uma melhor aceitação dos produtos e serviços da FSA no mercado, uma vez que os clientes externos poderão observar e comprovar a aplicabilidade e qualidade dos produtos e serviços oferecidos pela FSA já em uso na instituição (OLIVEIRA e NETO, 2003; BORGES, CARVALHO e MORAES, 2012; BRITO, SILVA e CABRAL, 2013; LEITE e LUCRÉDIO, 2014).

Outra clara tendência que pode ser observada é a que aponta para a adoção por FSA de metodologias de desenvolvimento de software derivadas do método ágil Scrum, seguindo uma tendência do mercado de Fábricas de Software convencionais. Pesquisas recentes indicam que esta metodologia é utilizada por cerca de 56% das empresas que utilizam metodologias ágeis, conforme pode ser observado no Gráfico 1 (VERSIONONE, 2015). Dentre as dez IES pesquisadas, seis relataram terem obtido sucesso ao adaptar e utilizar esta metodologia para operacionalizar a produção de software em suas fábricas. As demais instituições utilizam outras metodologias ou não informaram, conforme pode ser observado na Tabela 3.

O Gráfico 1 mostra como o Scrum domina o cenário de metodologias ágeis de desenvolvimento de software:

Gráfico 1 – Metodologias ágeis mais utilizadas atualmente no mundo



Fonte: Adaptado de (VERSIONONE, 2015)

Apesar de adotarem diferentes estruturas internas em suas Fábricas de Software; estarem inseridas em diferentes contextos e de possuírem variados níveis de maturidade em seus processos como apresentado na Tabela 3, as instituições analisadas relatam por unanimidade a ocorrência de uma relativa satisfação em relação aos resultados obtidos após a implementação de suas referidas FSA. Os artigos analisados e respostas obtidas apontam que de maneira geral, as FSA, após passarem pela fase de implementação, atendem às expectativas iniciais de suas instituições, apesar das dificuldades encontradas durante o processo de implementação das mesmas. Tanto as expectativas quanto as dificuldades relatadas são apresentadas em detalhes nos tópicos 4.2 e 4.3, respectivamente.

Maiores detalhes, inclusive aspectos técnicos relacionados à experiência de cada IES com o conceito de Fábrica de Software podem ser encontrados no Apêndice B.

A Tabela 3 apresenta as IES analisadas por este estudo, juntamente com o ano de inauguração de suas FSA, o modelo de negócio de cada uma (privada ou pública), o Estado em que se localizam, a estrutura interna e a metodologia ou padrão adotado no processo das equipes de desenvolvimento:

Tabela 3 – Exemplos de Fábricas de Software em IES no Brasil

Instituição	Tipo	UF	Ano	Estrutura (divisão interna)	Metodologias de processo
Universidade Federal do Pará (PACHECO, 2008)	Pública	PR	2005	Informação indisponível	MPS.Br
Faculdade Lourenço Filho (http://www.flf.edu.br/fabrica/home/quemsomos , 2012)	Privada	CE	2012	Engenharia de Software Linguagens de Programação Ambientes e Redes	Scrum
Faculdade de Tecnologia de Jundiaí (OLIVEIRA e NETO, 2003)	Pública	SP	2001	Gestão de Projetos Fábrica Lógica Fábrica Física	CMM
Universidade Federal de São Carlos (LEITE e LUCRÉDIO, 2014)	Pública	SP	2012	Gerência de Projetos Desenvolvimento	Scrum
Universidade Federal de Pernambuco (SOARES, MARIZ, CAVALCANTI, RODRIGUES, BASTOS, NETO, ALMEIDA, PEREIRA, ARAÚJO e ALBUQUERQUE, 2007)	Pública	PE	2007	Comitê gestor Pré-venda Planejam. /Acompanhamento Desenvolvimento Pós-venda Configuração/Mudanças	Scrum
Instituto Federal de Goiás - Campus Inhumas (BRITO, SILVA e CABRAL, 2013)	Pública	GO	2010	Informação indisponível	Scrum
Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul – Campus Porto Alegre (BORGES, CARVALHO e MORAES, 2012)	Pública	RS	2010	Gerência de Projetos Desenvolvimento	Scrum XP
Universidade Federal de Lavras (AMÂNCIO, COSTA, CAMARGO e PENTEADO, 2009)	Pública	M G	2009	Gerência de Projetos Desenvolvimento	RUP
Universidade Estadual de Londrina (GAFFO, BARROS e BRANCHER, 2012)	Pública	PR	2008	Informação indisponível	MPS.Br
Universidade de Brasília - Faculdade do Gama (VERGARA, 2014)	Pública	DF	2011	Informação indisponível	Scrum

Fonte: Elaborado pelo autor.

4.2 MOTIVAÇÕES PARA A IMPLEMENTAÇÃO DE UMA FSA

A principal motivação apontada pelas IES pesquisadas para a implementação de uma Fábrica de Software é a necessidade de proporcionar experiência prática aos alunos das disciplinas relacionadas à Engenharia de Software. Tal necessidade, além de ser facilmente percebida pelos professores das disciplinas, também é uma determinação do Ministério da Educação (MEC) por meio das Diretrizes Curriculares de Cursos da Área de Computação e Informática (CEEInf, 1999), portanto, as instituições que melhor proporcionam um ambiente de aprendizado prático aos seus alunos tendem a serem melhor avaliadas pelo MEC e a cumprirem de forma satisfatória a missão de preparar bons profissionais para o mercado de trabalho.

O segundo motivo mais apontado pelas instituições pesquisadas para a implementação de uma FSA é a possibilidade de atender a demandas internas por softwares vindas dos diversos setores da instituição. Isso se deve principalmente ao fato de que com a FSA é possível atender a essas demandas com um custo mais baixo se comparado ao custo de contratação de empresas especializadas. Nas IES que utilizam sua FSA para este fim, as equipes da Fábrica ficam responsáveis não só pelo desenvolvimento das soluções sistêmicas, mas também pelo suporte técnico necessário para garantir o funcionamento adequado dos produtos utilizados. Outro fator que influencia as instituições a adotarem este modelo é o alto nível de personalização das ferramentas desenvolvidas, uma vez que são feitas sob medida para atender as necessidades locais, ao contrário das soluções de mercado, que não se adequam em sua plenitude a tais necessidades, pois são criadas para atender a um grande número de clientes, que possuem necessidades variadas.

A diminuição da evasão e da desistência dos alunos é outro fator que levam as IES a implementar Fábricas de Software segundo estudo realizado. As instituições relatam que ao criar um ambiente interativo para a prática dos conhecimentos adquiridos no curso, o interesse dos alunos tende a aumentar, diminuindo assim a taxa de desistência. Vale ressaltar também, que este é um fator particularmente importante em contextos de crise econômica, onde cada aluno é disputado de maneira mais agressiva pelas IES.

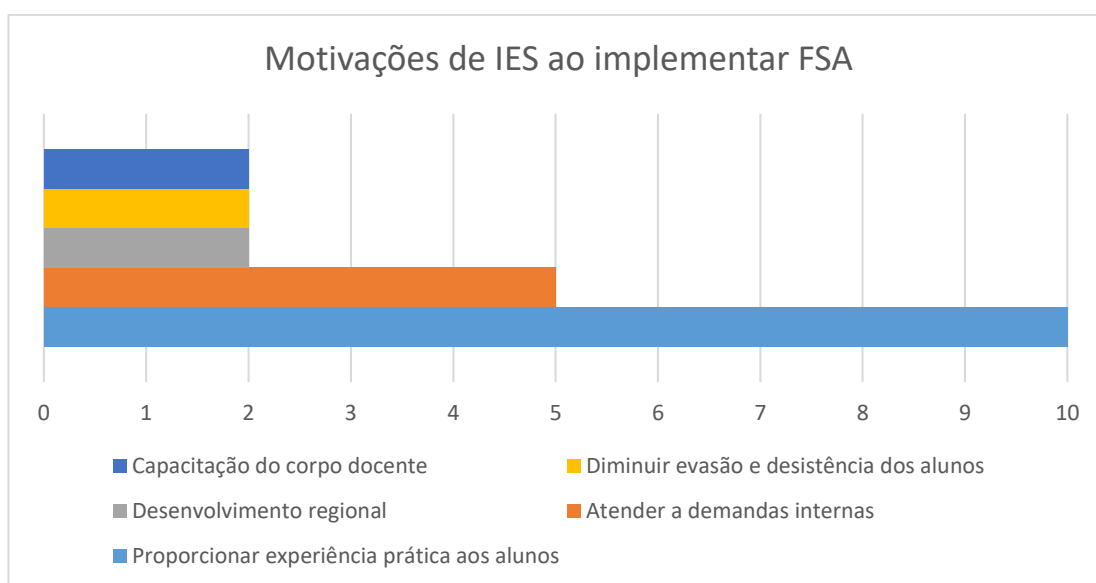
Foi relatado também por algumas instituições uma preocupação em relação ao desenvolvimento da região onde estão inseridas. Tais IES esperam que ao oferecerem serviços e produtos de custo reduzido e com qualidade por meio de suas FSA, além de gerarem recursos financeiros a serem revertidos para a própria instituição, podem também melhorar a produtividade das empresas da região, promovendo assim um melhor ambiente de negócios,

onde empregos e riqueza são gerados, atraindo novos habitantes e, conseqüentemente, mais alunos, em um ciclo que beneficia a todas partes envolvidas.

Por meio de suas FSA, algumas instituições buscam promover não somente a capacitação adequada de seus alunos, mas também de seus professores, uma vez que em uma FSA estes possuem papel central no acompanhamento e gerenciamento dos projetos em andamento, forçando com que se mantenham ainda mais atualizados em relação às práticas de mercado e novas tecnologias e ferramentas. Algumas instituições incluem nos custos de suas fábricas investimentos em treinamentos específicos para os docentes envolvidos, uma maneira também de aumentar a satisfação do profissional e conseqüentemente a retenção de talentos em seu quadro.

No Gráfico 2 aparece a quantidade de instituições pesquisadas que apontaram cada um dos fatores motivacionais mencionados anteriormente:

Gráfico 2 - Motivações de IES ao implementar FSA



Fonte: Elaborado pelo autor.

A Tabela 4 indica quais instituições apontaram cada um dos principais fatores identificados durante a pesquisa:

Tabela 4 – Fatores que influenciam para a implementação de Fábrica de Software em IES

Instituição	Proporcionar experiência prática aos alunos	Atender demandas internas da instituição	Desenvolvimento regional	Diminuir evasão e desistência dos alunos	Capacitação do corpo docente
Universidade Federal do Pará (PACHECO, 2008)	X			X	
Faculdade Lourenço Filho (http://www.flf.edu.br/fabrica/home/quemsomos , 2012)	X	X	X		
Faculdade de Tecnologia de Jundiá (OLIVEIRA e NETO, 2003)	X	X	X		X
Universidade Federal de São Carlos (LEITE e LUCRÉDIO, 2014)	X	X			
Universidade Federal de Pernambuco (SOARES, MARIZ, CAVALCANTI, RODRIGUES, BASTOS, NETO, ALMEIDA, PEREIRA, ARAÚJO e ALBUQUERQUE, 2007)	X				
Instituto Federal de Goiás - Campus Inhumas (BRITO, SILVA e CABRAL, 2013)	X	X			
Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul – Campus Porto Alegre (BORGES, CARVALHO e MORAES, 2012)	X	X		X	
Universidade Federal de Lavras (AMÂNCIO, COSTA, CAMARGO e PENTEADO, 2009)	X				
Universidade Estadual de Londrina (GAFFO, BARROS e BRANCHER, 2012)	X				X
Universidade de Brasília - Faculdade do Gama (VERGARA, 2014)	X				

Fonte: Elaborado pelo autor.

4.3 PRINCIPAIS DIFICULDADES ENFRENTADAS PELAS IES DURANTE O PROCESSO DE IMPLEMENTAÇÃO DE SUAS FSA

A análise das informações disponíveis sobre a experiência relatada por cada IES aponta algumas dificuldades relevantes encontradas durante o processo de implementação de suas referidas Fábricas de Software.

A dificuldade mais comumente relatada foi o desafio para conciliar os horários de disponibilidade de cada aluno membro de equipe de desenvolvimento. Isso porque nessas instituições as tarefas em um projeto não são distribuídas apenas entre estagiários de horário fixo, mas também entre alunos que trabalham de forma remota em horários alternativos, uma possibilidade pensada para viabilizar a participação daqueles que já exercem atividade remunerada em horário convencional. Para mitigar tal dificuldade, as instituições pesquisadas tiveram que adotar ferramentas e procedimentos que permitissem o trabalho remoto, incluindo ferramentas para reuniões virtuais e para controle de versão dos artefatos gerados, facilitando também a divisão de tarefas entre um número maior de participantes.

A segunda dificuldade mais mencionada foi a necessidade de adaptar certas características das metodologias de desenvolvimento adotadas. Isso se deve ao fato de que tais metodologias, como o Scrum, mais comumente utilizado, foram concebidas para serem aplicadas em um cenário onde a equipe de desenvolvimento é composta por profissionais experientes e cuja jornada de trabalho possui uma alta carga horária, geralmente em horário fixo, características que não estão presentes em equipes formadas por alunos de graduação, conforme já mencionado no parágrafo anterior. As principais adaptações relatadas estão relacionadas à frequência das reuniões periódicas e à quantidade de documentação gerada por cada projeto. Um exemplo mais específico desse tipo de adaptação é a mudança nos períodos padrões adotada para os ciclos de desenvolvimento, também chamados de Sprints, que normalmente são curtos, podendo ser de um a dois dias, mas que em FSA necessitam ser mais longos, como uma ou duas semanas. Mas vale ressaltar que isso não necessariamente significa um maior tempo de desenvolvimento, uma vez que isso vai depender também de outros fatores, como o tamanho das equipes, solidez da base de conhecimento da fábrica e capacidade técnica dos integrantes, incluindo o gerente do projeto, cargo ocupado por um professor em uma FSA.

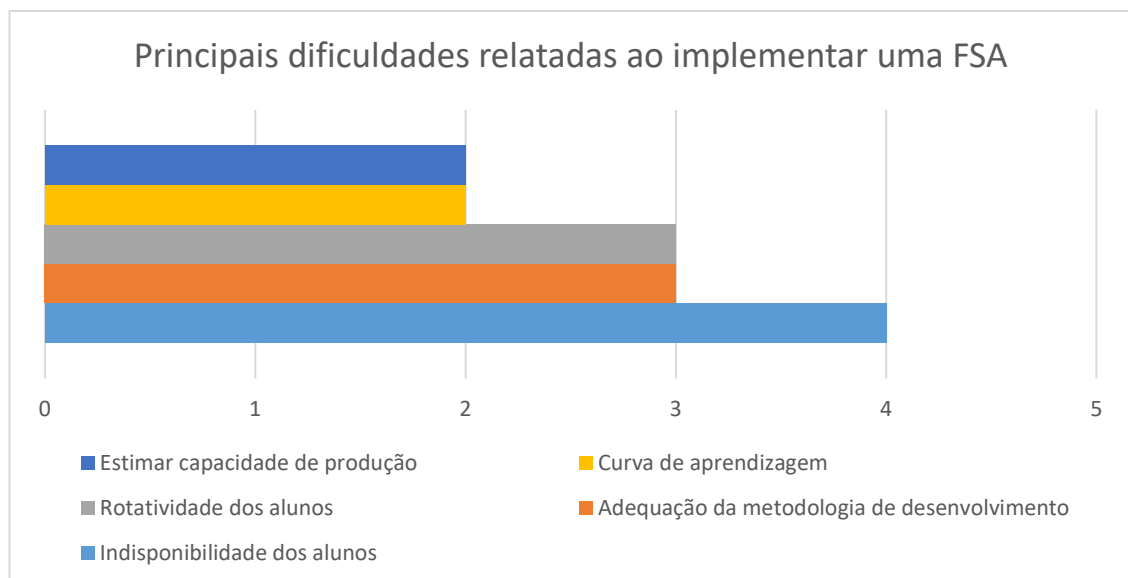
Devido ao fato de boa parte da mão de obra em uma Fábrica de Software Acadêmica ser composta por alunos de graduação, a rotatividade nas equipes é particularmente alta. Os principais motivos relatados para a saída de alunos foram: a conclusão do curso e propostas de emprego no mercado. Vale ressaltar também que por se tratar de alunos em formação, a curva de aprendizagem dos mesmos nas metodologias e ferramentas adotadas é maior que a

apresentada por profissionais experientes, outra dificuldade citada pelas instituições pesquisadas, fazendo com que o impacto da alta taxa de rotatividade seja ainda maior. Para minimizar o impacto desse problema e manter a constância das operações da Fábrica de Software, atendendo aos prazos estabelecidos, as instituições pesquisadas adotam medidas como a estruturação de uma base de conhecimento organizada, online, acessível mediante identificação, de maneira que novos integrantes das equipes possam se adequar mais rapidamente à metodologia utilizada e aprenderem com as lições ali armazenadas.

Segundo algumas IES pesquisadas, fazer estimativas em relação a capacidade de produção de cada equipe de desenvolvimento se mostrou uma tarefa complicada, principalmente durante a elaboração dos primeiros projetos da fábrica, pois não há base histórica que sirva de suporte para previsões de esforço necessário e também devido à alta taxa de rotatividade entre os membros das equipes, problema já mencionado anteriormente. Esta é uma dificuldade que tende a ser minimizada naturalmente com o passar do tempo, na medida em que as equipes adquirem experiência, o repositório de artefatos reaproveitáveis cresce e a base de conhecimento a respeito dos projetos anteriores permite gerar dados estatísticos de desempenho por tipo de componente criado.

O Gráfico 3 apresenta quantas instituições pesquisadas apontaram cada uma das dificuldades mencionadas anteriormente:

Gráfico 3 – Principais dificuldades relatadas ao implementar uma FSA



Fonte: Elaborado pelo autor.

A Tabela 5 apresenta as principais dificuldades relatadas por cada IES no que diz respeito à implementação e ao gerenciamento de suas referidas Fábricas de Software:

Tabela 5 – Principais dificuldades encontradas por cada IES durante o processo de implementação de Fábrica de Software

Instituição	Indisponibilidade dos alunos	Adequação da metodologia de desenvolvimento	Rotatividade dos alunos	Curva de aprendizagem	Estimar capacidade de produção
Universidade Federal do Pará (PACHECO, 2008)			X		
Faculdade Lourenço Filho (http://www.flf.edu.br/fabrica/home/que-msomos , 2012)			X		
Faculdade de Tecnologia de Jundiá (OLIVEIRA e NETO, 2003)				X	
Universidade Federal de São Carlos (LEITE e LUCRÉDIO, 2014)		X		X	
Universidade Federal de Pernambuco (SOARES, MARIZ, CAVALCANTI, RODRIGUES, BASTOS, NETO, ALMEIDA, PEREIRA, ARAÚJO e ALBUQUERQUE, 2007)	X	X			
Instituto Federal de Goiás - Campus Inhumas (BRITO, SILVA e CABRAL, 2013)	X	X			
Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul – Campus Porto Alegre (BORGES, CARVALHO e MORAES, 2012)					X
Universidade Federal de Lavras (AMÂNCIO, COSTA, CAMARGO e PENTEADO, 2009)	X		X		
Universidade Estadual de Londrina (GAFFO, BARROS e BRANCHER, 2012)					X
Universidade de Brasília - Faculdade do Gama (VERGARA, 2014)	X			X	

Fonte: Elaborado pelo autor.

5 PROPOSTA DE FÁBRICA DE SOFTWARE PARA A ASSOCIAÇÃO EDUCACIONAL DOM BOSCO (AEDB)

A proposta apresentada neste capítulo adequa-se às necessidades e particularidades da instituição e tem como base, além das experiências relatadas pelas demais Instituições de Ensino Superior estudadas, a ampla pesquisa documental e bibliográfica realizada. Primeiramente é apresentado um panorama do curso de Sistemas de Informação, de modo a contextualizar o ambiente em que está inserido o aluno participante da FS, em seguida, são apresentados os detalhes da proposta de implementação da FSA da AEDB e posteriormente, os resultados parciais obtidos.

5.1 O curso de Sistemas de Informação

O curso de Bacharelado em Sistemas de Informação (S.I) tem por objetivo a formação de profissionais para atuação focada no planejamento, na análise, na utilização e na avaliação de modernas tecnologias de informação aplicadas às áreas administrativas e industriais, em organizações públicas e privadas.

Este curso é definido pelo MEC (2000) como atividade meio. Isto quer dizer que o profissional não tem o planejamento do seu perfil voltado para a criação de novas tecnologias (como é o caso do perfil desejado para o egresso de Ciência da Computação), mas para a aplicação das tecnologias disponíveis na obtenção de resultados para as empresas.

O bacharel em Sistemas de Informação é o profissional responsável por planejar, manipular e administrar os fluxos de informação, gerados e distribuídos por redes de computadores, dentro de uma organização. Seu foco é analisar os problemas, apontar suas causas raízes e, fazendo uso de recursos tecnológicos, eliminá-los ou diminuí-los, aumentando com isso o controle, produtividade e/ou reduzindo as perdas e prejuízos que uma empresa possa vir a ter pela má administração/utilização de seus recursos. Esse profissional planeja e orienta o processamento, o armazenamento e a recuperação de informações e o acesso de usuários a elas. Desenvolve novos sistemas, administra banco de dados e redes corporativas, cria programas e soluções que viabilizam a tramitação das informações por estas redes utilizando a tecnologia adequada para os diversos tipos de problemas que uma corporação possa vir a ter.

No curso de S.I da AEDB, os alunos têm à sua disposição laboratórios de informática com equipamentos modernos e softwares de última geração, necessários para um aprendizado prático e atualizado. Tais laboratórios são utilizados durante as aulas, que ocorrem no período

noturno e passarão a ser utilizados pela FSA durante o dia, evitando que fiquem ociosos e valorizando ainda mais os ativos da instituição.

5.2 A PROPOSTA

Nesta sessão são apresentados os detalhes da proposta de FSA que foi elaborada, e apresentada à direção da AEDB e à coordenação do curso de Sistemas de Informação da mesma, que a aprovaram sem ressalvas. Inicialmente é abordada a composição das equipes de projeto da fábrica, em seguida são explicados os detalhes a respeito do ciclo de permanência dos alunos na FSA. Posteriormente são apresentadas metodologias e procedimentos estabelecidos para o bom funcionamento da unidade de produção e, por último, é mostrado o cronograma e o planejamento da FSA para os próximos anos, assim como as etapas já cumpridas nos anos anteriores.

5.2.1 Membros da Fábrica de Software da AEDB

Conforme metodologia de desenvolvimento recomendada pelas instituições pesquisadas e escolhida para a FS da AEDB, o Scrum, haverá sempre três papéis em cada projeto da FSA:

- **Project Owner:** responsável pela macro gerência, ou seja, por gerenciar “o que” deve ser feito. Na FSA da AEDB este papel poderá ser desempenhado por: um representante do setor onde o projeto será utilizado depois de finalizado, caso seja produto para uso interno; um representante de um cliente externo à instituição, que domine os requisitos do produto a ser desenvolvido ou um professor analista, com ampla visão de negócio, responsável por realizar o intermédio entre o cliente e a equipe técnica.
- **Scrum Master:** responsável por gerenciar o processo, ou seja, garantir que a metodologia do Scrum está sendo posta em prática. Este papel será exercido por professores do curso de Sistemas de Informação da AEDB que possuem o conhecimento técnico necessário.
- **Equipe de desenvolvimento:** responsável pela micro gerência, ou seja, responsáveis por executar o trabalho operacional seguindo procedimentos e métodos estabelecidos, papel que será desempenhado pelos alunos do curso de Sistemas de Informação.

As equipes de desenvolvimento da FSA da AEDB poderão ser compostas por três diferentes perfis de alunos, são eles:

- Alunos que participam dos projetos de maneira não presencial na maioria do tempo, tendo flexibilidade de horário e que, por isso, fazem uso de ferramentas de trabalho remoto. Tais alunos tem como atrativo para participar da FS a possibilidade de conseguir horas complementares exigidas pelo curso, além do aprendizado prático a respeito do conteúdo aprendido em sala de aula.
- Alunos estagiários da instituição, com carga horária e período de trabalho bem definidos. Estes alunos têm como benefício pela participação nos projetos, além do aprendizado prático, o cumprimento do estágio obrigatório exigido pelo curso e, no caso da AEDB, a isenção do pagamento pelos estudos, por meio de bolsa integral acrescida de uma remuneração extra.
- Alunos do último ano do curso, que optem por incluir seu trabalho de conclusão na lista de projetos da FS, seguindo assim as metodologias e procedimentos por ela estabelecidos. Para estes alunos, o atrativo adicional para incluir seus projetos no portfólio da fábrica é o fato de poderem utilizar da estrutura física e lógica da FS para o desenvolvimento de seu trabalho de conclusão do curso, dessa maneira adicionando robustez e profissionalismo ao mesmo.

Os dois primeiros grupos de alunos citados acima podem cooperar simultaneamente no mesmo projeto, fazendo com que a dinâmica do gerenciamento destas equipes demande uma atenção especial daqueles que exercem esta função, no caso, os professores do curso de Sistemas de Informação.

A Tabela 6 apresenta os atrativos para os diferentes perfis de alunos da FS da AEDB:

Tabela 6 – Vantagens da participação na FSA da AEDB por perfil de aluno:

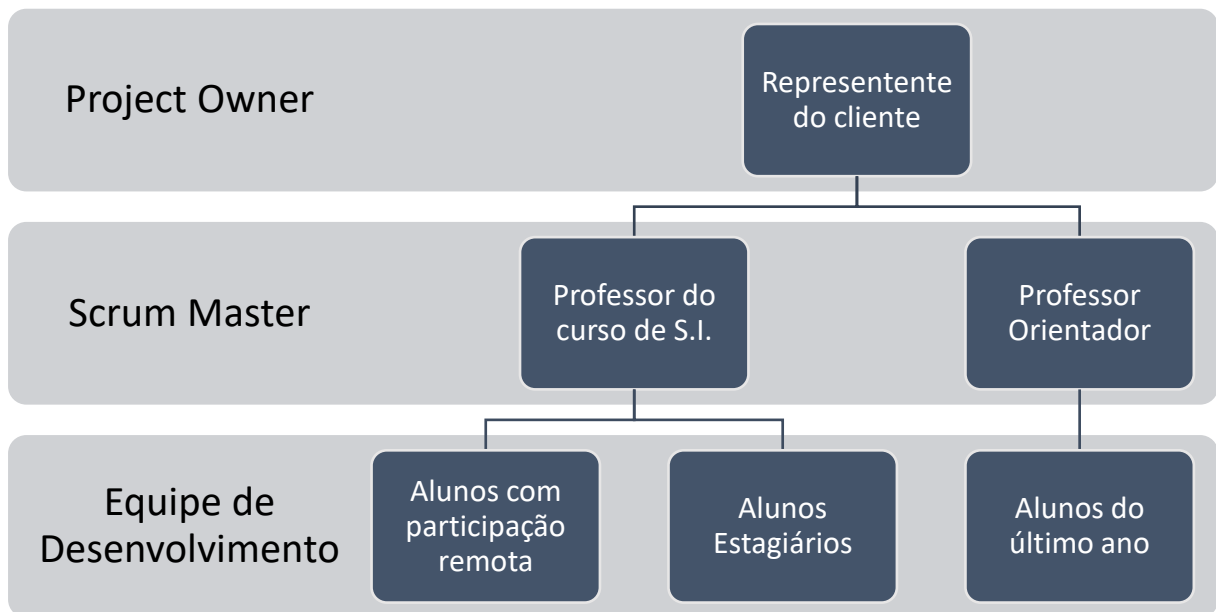
Perfil do aluno	Aprendizado prático	Horas complementares	Estágio obrigatório	Profissionalizar o projeto
Participam remotamente	X	X		
Estagiários	X		X	
Formandos	X			X

Fonte: Elaborado pelo autor.

No caso dos alunos do último ano do curso que tiverem seus projetos incluídos no portfólio da FS, estes terão que desenvolvê-lo sem a ajuda técnica dos demais alunos e tendo seu professor orientador exercendo o papel de gerente de projeto, conforme demanda a metodologia de desenvolvimento adotada pela fábrica. Após o término do projeto, estando os alunos formados, os mesmos terão duas opções para que seu produto seja continuado: assumirem a responsabilidade pelo mesmo, ou seja, empreendendo, ou deixa-lo a cargo das equipes da FS para que o mantenham disponível para ser oferecido a futuros clientes interessados. Para que esta segunda opção seja possível, os alunos terão de assinar um termo no início do projeto se comprometendo a utilizarem as tecnologias e metodologias da fábrica de software, de modo a possibilitar a manutenção do produto pela mesma posteriormente.

A figura abaixo ajuda a entender a distribuição dos membros das equipes de projeto da FSA da AEDB:

Figura 4 - Membros das equipes de projeto FSA da AEDB



Fonte: Elaborado pelo autor.

A fábrica poderá ser composta por diversas equipes, sendo cada equipe detentora de suas próprias células de produção, distribuídas levando-se em consideração a competência de seus participantes. As primeiras equipes deverão trabalhar em projetos que atendam a demandas internas da instituição, de modo a proporcionar amadurecimento dos processos estabelecidos antes de lidar com clientes externos, estratégia seguida por diversas instituições que obtiveram sucesso em projetos semelhantes, conforme já apresentado no Tópico 4.1.

5.2.2 Ciclo de permanência dos alunos na FSA

Diferente da maioria das faculdades do Brasil, onde os cursos são divididos em semestres, na AEDB eles são divididos em anos, sendo cada ano dividido em quatro bimestres. O curso de Sistemas de Informação é composto por quatro períodos de um ano e baseando-se nisso, esta proposta prevê um ciclo de permanência dos estudantes na fábrica de até três anos, sendo cada ano dedicado a uma diferente etapa do processo de aprendizado.

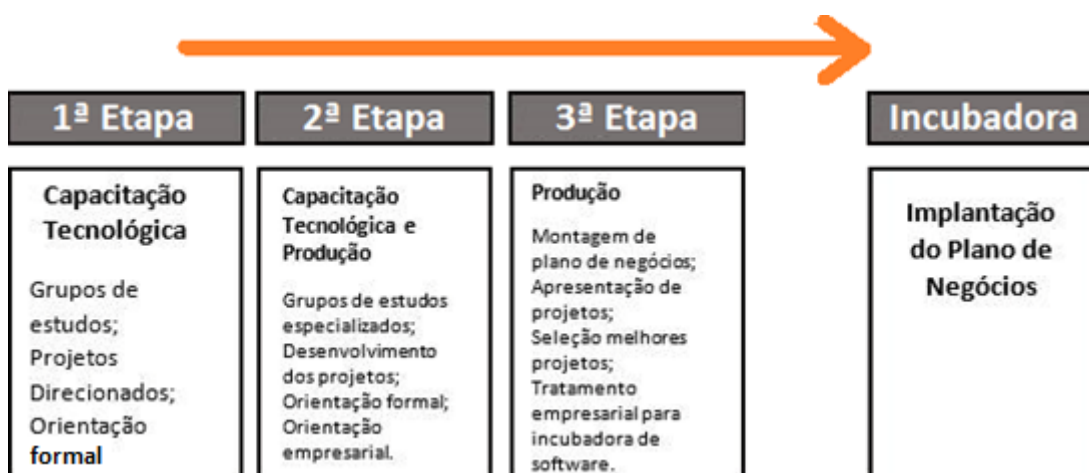
A primeira etapa, de duração de um ano, é destinada preferencialmente aos alunos do segundo ano do curso e visa promover a capacitação tecnológica, etapa que será realizada por meio da formação de grupos de estudos com orientação personalizada.

Na segunda etapa, destinada preferencialmente aos alunos do terceiro ano do curso, os próprios estudantes terão liberdade para apresentar suas propostas de projetos ou serem recrutados para compor equipes de projetos já em andamento. Neste período os estudantes receberão, além da contínua capacitação tecnológica, orientação para a montagem de planos de negócios e assumirão maior responsabilidade na equipe em que estiverem inseridos.

A terceira e última fase, que visa a participação preferencial dos alunos do quarto e último ano do curso, busca estimular os estudantes a fazerem uso dos ensinamentos das fases anteriores para apresentarem seus próprios planos de negócios, que poderão vir a serem hospedados em uma incubadora de empresas de tecnologia, a qual será proposto que funcione na própria AEDB. Um dos principais objetivos desta última fase é estimular o senso empreendedor dos alunos e transmitir a eles melhores noções de atendimento ao cliente e visão de mercado.

A Figura 5 apresenta o ciclo de permanência proposto:

Figura 5 - Ciclo máximo de permanência dos alunos na Fábrica de Software



Fonte: Elaborado pelo autor.

5.2.3 Metodologias e procedimentos

A metodologia de desenvolvimento a ser adotada pelo corpo técnico envolvido nos projetos, Scrum (Apêndice B), precisou ser adaptada para se adequar a realidade da instituição, as principais mudanças estão relacionadas aos períodos padrões adotados para os ciclos de desenvolvimento, também chamados de Sprints, e também na forma de comunicação entre os membros das equipes, fazendo uso de ferramentas de comunicação instantânea e trabalho remoto, conforme fizeram outras IES já mencionadas no Capítulo 4.

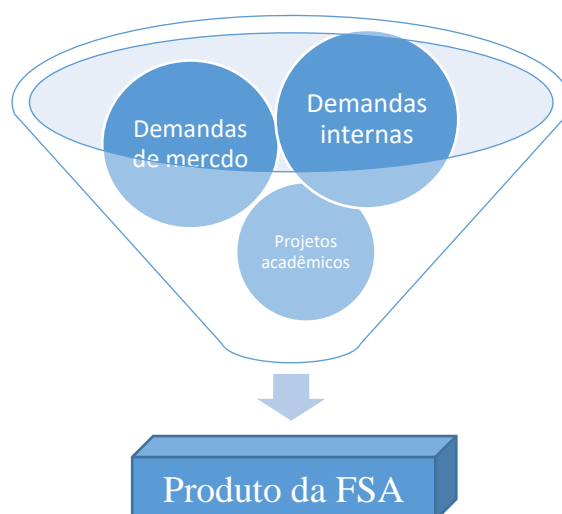
Os projetos da FSA da AEDB poderão ser provenientes de diferentes fontes, tais como:

- Demandas internas à instituição: projetos que visam atender necessidades de setores da AEDB que buscam adquirir soluções sistêmicas para seus problemas operacionais.
- Demandas de outras empresas ou da sociedade: projetos originados de demandas de mercado, os quais podem gerar recursos financeiros para a instituição, a ser revertido para melhorias na própria FSA.
- Projetos meramente acadêmicos: provenientes de ideias apresentadas pelo corpo acadêmico da AEDB, que podem ou não vir atender a demandas existentes, ou seja, podem ser projetos que visam apenas testar conceitos, novas tecnologias ou atender a demandas que ainda não foram formalmente manifestadas.

Projetos originados de qualquer uma das três origens mencionadas acima, podem vir a ser desenvolvidos tanto no formato de projetos de conclusão de curso quando no formato convencional, conforme já explicado no Tópico 5.2.1.

A imagem abaixo ilustra as possíveis origens dos projetos da FSA da AEDB:

Figura 6 – Origens dos projetos da FSA da AEDB



Fonte: Elaborado pelo autor.

Os diversos artefatos ou subprodutos gerados durante o desenvolvimento serão armazenados em um banco de componentes, onde serão devidamente catalogados de acordo com suas características e funcionalidades, permitindo que tais elementos possam ser utilizados em novos projetos. O reaproveitamento de código irá reduzir o tempo de desenvolvimento dos projetos e melhorar as estimativas de capacidade. Os objetos deverão ser armazenados em uma base de dados de acesso restrito, hospedada na nuvem privada da instituição.

Estão sendo avaliadas possíveis parcerias entre a recém-criada Fábrica de Software da instituição e empresas de desenvolvimento de software da região. Tais parcerias tem se mostrado vantajosas para ambos os lados, uma vez que a Fábrica poderá fazer uso, principalmente em seu período inicial, de todo o conhecimento operacional e técnico já acumulado por essas empresas, no que tange à gerenciamento de projetos, uso de ferramentas específicas e do próprio mercado, enquanto às empresas poderão se beneficiar de todo o conhecimento e ambiente acadêmico da instituição de ensino. Uma das empresas que já se mostrou interessada é a Empório High-Tech (<http://www.emporiohightech.com.br/>), também com sede em Resende-RJ. Um bom exemplo de parceria semelhante à que está sendo proposta é a existente entre a empresa Sofhar (<http://www.sofhar.com.br/>) e o Parque Tecnológico da Pontifícia Universidade Católica do Paraná (PUCPR), que em 2013 lançaram em conjunto um sistema de BI (Business Intelligence), onde a empresa entrou com o know-how de negócios e a universidade com o conhecimento científico e mão de obra (MELLO, 2013).

Há também a preocupação de que a metodologia de gerenciamento de equipes e de projetos adotada na fábrica não seja demasiadamente burocrática, permitindo certo nível de flexibilidade nos processos, ideia fundamentada por meio da Metodologia de Gerenciamento de Projetos do Sistema de Administração de Recursos de Tecnologia da Informação do Governo Federal (SISP, 2011). Segundo o SISP, o gerenciamento do projeto não deve ter esforço maior do que a própria execução. Ou seja, a quantidade de artefatos gerados em um determinado projeto deve ser adequada ao tamanho do projeto e ao esforço do seu gerenciamento.

A FSA da AEDB fará uso de um sistema de controle de chamados implementado recentemente na instituição, facilitando com isso a manutenção dos sistemas disponibilizados e garantindo a devida padronização dos processos da área de suporte.

5.2.4 Impacto acadêmico do projeto

A literatura analisada, que relata a experiência com Fábricas de Software de outras IES, aponta para a necessidade de garantir que as ementas das disciplinas do curso de tecnologia contemplem o conhecimento básico necessário para que os alunos tenham condições de se adequar às técnicas, ferramentas e métodos adotados pela fábrica. Desta forma, foi feita uma revisão de todas as ementas do curso de Sistemas de Informação da AEDB, visando identificar a necessidade de adaptações em função desta nova demanda. A análise resultou em alterações nas ementas das seguintes disciplinas:

- Gerência de Projetos – Inclusão de aspectos relacionados especificamente a projetos de desenvolvimento de software utilizando métodos ágeis, como o Scrum. Passou a prever a realização de simulações práticas para familiarizar os alunos com os procedimentos deste novo paradigma.
- Engenharia de Software – Ementa alterada para contemplar explicações a respeito da documentação gerada em projetos de uma Fábrica de Software, assim como estimular por meio de exercícios em aula e competições entre equipes a agilidade dos alunos.
- Modelagem de Sistemas – Melhor explicação a respeito da fase de testes de um sistema, para que os alunos entendam a importância desta etapa e conheçam os principais modelos de documentos e práticas relacionados a ela.
- Criação de uma nova disciplina denominada “Metodologias de Desenvolvimento de Sistemas”, para contemplar em detalhes cada uma das principais metodologias de desenvolvimento ágil da atualidade, como XP, Scrum, RUP, assim como abordar as diferentes certificações de processos de software existentes, como CMMI e o MPS.BR. A ementa elaborada para esta disciplina encontra-se no Anexo A.

Além das mudanças em ementas, alguns professores passaram a considerar a participação dos alunos nas atividades da Fábrica de Software como parte do sistema de avaliação de suas disciplinas, o que motiva os alunos a buscarem participar e dá oportunidade ao professor de reforçar o conteúdo ministrado com atividades práticas reais e não apenas hipotéticas.

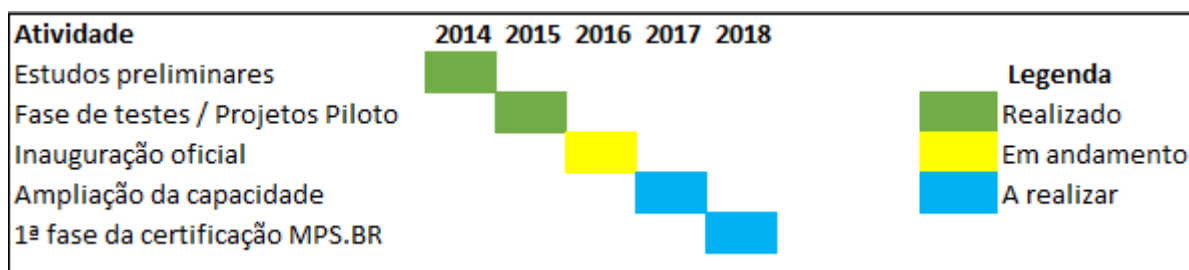
Com estas mudanças, que entraram em vigor em 2016, o curso de Sistemas de Informação da AEDB passou a preparar melhor seus alunos para atuarem em Fábrica de Software, atendendo às demandas do mercado por mão de obra especializada para este cenário e também contribuindo para uma menor curva de aprendizado dos alunos envolvidos na FSA da própria instituição.

5.2.5 Cronograma e planejamento

A Fábrica de Software da AEDB passou por testes preliminares em 2015 e entrou em funcionamento oficialmente ainda no primeiro semestre do ano letivo de 2016. Inicialmente estava prevista a contratação de alunos estagiários já em 2016 para comporem os quadros da FSA, no entanto, devido aos impactos da atual crise econômica tais contratações tiveram de ser adiadas para o próximo ano, de modo que em 2016 a fábrica funcionará apenas com grupos alunos trabalhando remotamente e também pelos formandos, além de grupos de formandos do curso de S.I, conforme possibilidade levantada no Tópico 5.2.1 desde documento. Para o ano de 2017 é previsto que a fábrica seja patrocinada por meio de uma linha de pesquisa criada especificamente para esse propósito.

Em 2018, o nível de maturidade dos processos da FSA será analisado por uma comissão interna, que irá determinar o esforço necessário para que a instituição possa buscar o primeiro nível de certificação do MPS.BR, certificação que melhor se adequa a unidades de produção de pequeno porte, conforme explicado no Apêndice C, desta forma melhorando a confiança dos clientes na qualidade dos produtos oferecidos. A certificação é recomendada para este tipo de ramo de atuação, conforme indicado tanto pela revisão bibliográfica, quanto pelas Instituições de Ensino Superior pesquisadas.

Figura 7 – Cronograma macro da FSA da AEDB



Fonte: Elaborado pelo autor

5.3 RESULTADOS PARCIAIS

Conforme já mencionado, no ano de 2015 foram realizados alguns testes de conceito, onde foram aplicadas as metodologias e procedimentos descritos no Tópico 5.2. Neste período a fábrica contou com a participação de um professor especialista, um aluno estagiário, um profissional de TI da instituição e grupos de alunos trabalhando remotamente. Este grupo inicial

pôde gerar nesse período alguns produtos piloto, que conforme estratégia já explicada, visaram atender a demandas internas da instituição.

O processo de recrutamento dos alunos que iriam trabalhar nestes projetos de forma remota se deu da seguinte maneira: foram divulgadas nas turmas de 2º, 3º e 4º ano do curso de Sistemas de Informação a oportunidade de 3 alunos participarem dos projetos piloto da Fábrica de Software, em seguida foi disponibilizado um formulário online de inscrição dos interessados. Ao todo, 7 alunos se inscreveram para participar do processo seletivo, sendo que destes, 3 foram selecionados, após passarem também por uma breve entrevista.

O formulário elaborado exigia a realização prévia pelos candidatos de uma lista de 4 cursos online gratuitos sobre programação, maneira encontrada para nivelar a equipe quanto ao mínimo de conhecimento prévio necessário, visto que as oportunidades foram abertas para alunos em diferentes momentos do curso. O formulário também visava medir o grau de experiência dos interessados, por meio de perguntas específicas com esse propósito. O formulário na íntegra encontra-se no Apêndice E.

Nos tópicos a seguir serão apresentados os dois projetos pilotos elaborados pela FSA da AEDB entre outubro de 2015 e janeiro de 2016. O primeiro deles, foi desenvolvido para a área de trabalho do sistema operacional Windows, utilizando a linguagem de programação C#.NET. Já o segundo projeto baseou-se em desenvolvimento Web, mas para ser utilizado na intranet da instituição, desenvolvido na linguagem de programação PHP para ser executado em navegadores de internet.

Após a implementação de cada solução, foi realizada uma pesquisa com o respectivo Project Owner, por meio de um formulário online, onde os mesmos puderam, além de demonstrar seu grau de satisfação, indicar possíveis pontos de melhoria ou fazer elogios. O formulário utilizado encontra-se no Apêndice F.

5.3.1 Projeto de interface entre os sistemas Gennera e TopAcesso

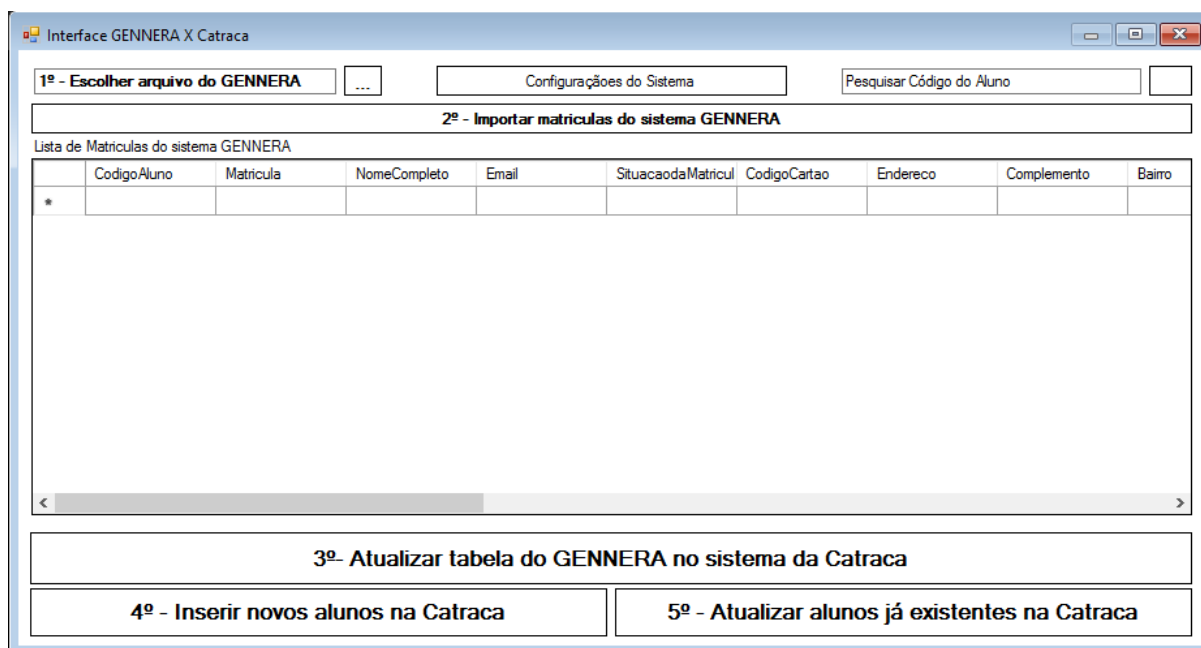
A AEDB hoje faz uso de diferentes softwares nas suas operações do dia a dia. Cada uma dessas ferramentas visa atender às necessidades de um setor específico da instituição. O primeiro projeto elaborado pela FSA procurou estabelecer comunicação entre o sistema responsável pelo controle de acesso dos alunos na instituição, o gerenciador de catracas chamado TopAcesso, e o sistema de gerenciamento acadêmico utilizado pela AEDB, chamado Gennera.

O problema a ser resolvido consistia na dificuldade em manter atualizada no sistema da catraca (TopAcesso) a lista dos alunos autorizados, informação proveniente do sistema acadêmico (Gennera), de modo que cada atualização de status da matrícula de um aluno no sistema acadêmico demandava uma atualização manual no sistema de controle de acesso à Faculdade.

Nesse projeto, a equipe de desenvolvimento foi composta por um aluno estagiário e 3 alunos trabalhando remotamente, tendo um professor do curso de Sistemas de Informação, como Scrum Master e o chefe do CPD (Centro de processamento de Dados) da AEDB como Project Owner.

O projeto teve duração de dois meses, sendo entregue no mês de dezembro. Hoje a solução desenvolvida encontra-se em pleno funcionamento, fazendo com o pressionar de alguns botões a atualização de todos os alunos de uma só vez no sistema de controle de acesso, de acordo com informações do sistema acadêmico.

Figura 8 – Tela principal do primeiro software desenvolvido pela FSA da AEDB



Fonte: Elaborado pelo autor.

O cliente, que nesse caso é o próprio Project Owner, ao responder a pesquisa realizada após a implementação do sistema, indicou grau máximo de satisfação com o produto e deixou o seguinte comentário:

“Poderia rodar automaticamente, sem precisar disparar manualmente, mas em comparação a como era feito, está muito melhor. Atendeu ao que foi proposto.”

Marcelo Oliveira – Coordenador do CPD da AEDB

5.3.2 Projeto de interface entre os sistemas Gennera e Moodle

O segundo projeto desenvolvido pela FSA da AEDB consistiu em criar uma interface de comunicação entre o sistema responsável pelo gerenciamento das disciplinas de Educação à Distância (EAD) e o sistema principal de gerenciamento acadêmico (Gennera).

Este projeto procurou resolver o fato de os professores das disciplinas EAD terem trabalho dobrado ao lançar as notas das atividades realizadas, uma vez no sistema Moodle e outra no sistema Gennera. Portanto, o produto a ser desenvolvido tinha como objetivo possibilitar a importação das notas de um sistema para o outro, eliminando assim a chance de erro humano e facilitando o dia a dia tanto dos professores quanto dos alunos, que tinham suas notas disponíveis para consulta mais rapidamente.

O projeto teve como equipe de desenvolvimento três alunos voluntários trabalhando remotamente, como Scrum Master, um analista do departamento de TI da AEDB e como Project Manager um professor.

Após dois meses de desenvolvimento, de dezembro a janeiro, o projeto encontra-se pronto para ser utilizado em ambiente de produção, o que ocorrerá ao final do primeiro bimestre do atual ano letivo, já tendo passado por todos os testes necessários, tendo sido apresentado à direção e à secretaria da instituição.

Figura 9 – Tela principal do segundo software desenvolvido pela FSA da AEDB

Fonte: Elaborado pelo autor.

O cliente, que nesse caso é também o próprio Project Owner, ao responder a pesquisa realizada após a implementação do sistema, também indicou grau máximo de satisfação com o produto, assim como ocorreu no projeto mencionado anteriormente. Além disso, ele deixou o seguinte comentário ao responder a pesquisa:

“Eliminou boa parte do trabalho manual que tínhamos no departamento, assim como as chances de erro de digitação e facilitou a vida principalmente dos professores. Parabéns.”

Pedro Ramirez – Coordenador de EAD da AEDB

6 CONCLUSÃO

Este trabalho analisou aspectos relacionados à implementação de Fábricas de Software em instituições de ensino superior no Brasil. A partir da pesquisa documental, de questionários e de uma ampla fundamentação teórica, foi possível identificar fatores que motivam as IES a implementar uma Fábrica de Software Acadêmica, assim como as principais dificuldades encontradas e demais aspectos relevantes. O estudo apresentou um comparativo entre as instituições analisadas, contribuindo para um melhor entendimento do cenário das FSA no Brasil e auxiliando futuras IES que desejam implementar projetos semelhantes para que se atentem aos fatores de sucesso e fracasso determinantes.

Baseando-se nas informações obtidas, foi elaborada uma proposta de Fábrica de Software para uma IES localizada no sul do estado do Rio de Janeiro, a AEDB, que utilizou os dados apresentados nesta pesquisa para se atentar a aspectos relevantes a serem considerados durante a execução do projeto, que já foi implementado e vem atendendo às expectativas da instituição, que inclusive planeja ampliar nos próximos anos a capacidade de sua recém instalada FSA, possibilitando assim a participação de um número cada vez maior de alunos e professores.

A implementação do projeto de FSA na AEDB demandou alterações na estrutura curricular de algumas disciplinas do curso de Sistemas de Informação, que passaram a abordar temas chave de maneira mais realista, através de metodologias ativas e projetos que seguem as atuais boas práticas do mercado.

Observa-se, portanto, que tanto os resultados parciais obtidos pelo projeto na AEDB quanto os relatos das IES pesquisadas apontam para a conclusão de que a implementação de uma Fábrica de Software em um ambiente acadêmico tende a contribuir para a formação adequada dos alunos dos cursos de tecnologia, proporcionando a eles a experiência prática necessária, além de trazer vantagens para toda a comunidade ao redor e para a própria instituição, que além de utilizar dessa estrutura para atender demandas internas e externas, tende a obter também um melhor reconhecimento da qualidade de seus cursos da área de TI.

6.1 SUGESTÃO PARA TRABALHOS FUTUROS

Apresenta-se como oportunidade de pesquisa, um estudo comparativo também entre as Fábricas de Software existentes em IES no exterior, de modo a identificar aspectos ainda não abordados no âmbito nacional, como novas metodologias e processos.

REFERÊNCIAS

AMÂNCIO, S. F.; COSTA, H. A. X.; CAMARGO, V. V.; PENTEADO, R. A. D. Gerência de Recursos Humanos para uma Fábrica de Software de Pequeno Porte. Lavras – MG, Universidade Federal de Lavras, 2009.

ARMOUR, F.; MILLER, G.. Advanced use case modeling: software systems. Pearson Education, 2000.

AUDY, J. Effectiveness of fiscal incentives to attract IT investments: A brazilian case. The Electronic Journal on Information Systems in Developing Countries, v. 15, n. 2, 2003. Disponível em: <www.ejisdc.org>. Acesso em: 15 jan.2015.

BASTOS, V. D. 2000-2010: Uma década de apoio federal à inovação no Brasil. Revista do BNDES, Rio de Janeiro, n. 37, p. 127-175, 2012.

BOENTE, A. N. P.; OLIVEIRA, F. S. G.; ALVES, J. C. N. RUP como Metodologia de Desenvolvimento de Software para Obtenção da Qualidade de Software. Resende – RJ, SEGeT – Simpósio de Excelência em Gestão e Tecnologia, 2005.

BORGES, K. S.; CARVALHO, T. P.; MORAES, M. A. C. Programa de Extensão “Fábrica de Software Acadêmica”: contribuindo para a formação profissional na área da Informática. Porto Alegre – RS, Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul (IFRS), 2012.

BRASIL, ISD. Fábrica de software da IBM Brasil conquista o nível 5 do CMMI. Disponível em: <<http://www.isdbrasil.com.br/imprensa.php?ID=28>>. Acesso em: 07 mar. 2016.

BRASIL. Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação. Metodologia de Gerenciamento de Projetos do SISP. Brasília, MP, 2011.

BRITO, M. C. A.; SILVA, F. P.; CABRAL, E. P. Elaboração de uma metodologia de desenvolvimento de software para a fábrica de software de uma instituição de ensino. Revista Brasileira de Informática na Educação, Vol. 21, Núm. 2, 2013.

BRITO, R.; FERREIRA, P.; SILVA, K.; BURÉGIO, V.; LEITE, I. Uma experiência na implantação de processo em uma fábrica de software livre. Recife – PE, Universidade Federal de Pernambuco, 2004.

CAMPELO, R. F.; SILVA, L. A. Identificar Etapas Convergentes entre o Método Cleanroom e a Metodologia Ágil Scrum. Universidade Municipal de São Caetano do Sul, 2011.

CARMEL, E. The new software exporting nations: Success factors. The Electronic Journal on Information Systems in Developing Countries, v. 13, n. 4, 2003. Disponível em: <www.ejisd.org>. Acesso em: 10 jan.2015.

CARVALHO, B. V.; MELLO, C. H. P. Aplicação do método ágil Scrum no desenvolvimento de produtos de softwares em uma empresa de base tecnológica. Gestão e Produção, São Carlos, v. 19, n. 3, p. 557- 573, 2012.

CARVALHO FILHO, H. C. Fábrica de Software: Um Estudo de Caso, Sob a Ótica da Flexibilização Organizacional e das Relações de Trabalho. Rio de Janeiro, Universidade Federal do Rio de Janeiro, 2008.

CASTOR, E. M. Fábrica de Software: Passado, Presente e Futuro. UNIBRATEC – União dos Institutos Brasileiros de Tecnologia. Recife - PE. 2006.

CASTRO, E. B. Um perfil do profissional de computação e informática. Dissertação (Mestrado) — Universidade Federal da Paraíba, João Pessoa, 2001.

CEEInf. Diretrizes Curriculares de Cursos da Área de Computação e Informática. 1999. Disponível em: <<http://www.inf.ufrgs.br/ecp/docs/diretriz.pdf>>. Acesso em: 12 out 2015.

CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. CMMI: Guidelines for process integration and product improvement. Boston, Massachusetts, USA: Addison-Wesley, 2003. (SEI Series in Software Engineering).

COSTA, C. M. Plano pedagógico para cursos de bacharelado em sistemas de informação. In: CURSO DE QUALIDADE - PLANOS PEDAGÓGICOS NA ÁREA DE COMPUTAÇÃO E INFORMÁTICA, 3. Fortaleza: Sociedade Brasileira de Computação, 2001.

CRUZ, S. M. S. QUISPE, F.; SUCUPIRA, G.; LEONARDO, J.; MATHEUS, L.; MONSORES, L.; YAGUI, M.; CHAN, V.; LIMA, Y. Relato de um Experimento Piloto de uma Fábrica de Software Baseada em Métodos Ágeis. Recife – PE, Universidade Federal de Pernambuco, 2013.

CUSUMANO, Michael A. The "factory" approach to large-scale software development: implications for strategy, technology, and structure. 1987.

DE GODOY, J. M. T. O mundo fabril nas concepções de Taylor, Fayol e Ford. Esboços-Revista do Programa de Pós-Graduação em História da UFSC, v. 17, n. 24, p. 37-70, 2011.

DIGITAL, P. Accenture abre fábrica de software em Recife. Disponível em: <<http://www2.portodigital.org/portodigital/imprensa/entrevistas/40508;56215;0802;4640;18164.asp>>. Acesso em: 07 mar. 2016.

DORIGAN, J. A. Gerenciamento de Requisitos: Um Comparativo entre Metodologias Tradicionais e Ágeis sob a ótica dos Modelos de Qualidade. 2010. 59 p. Dissertação (Graduação em Ciência da Computação) – Departamento de Computação, Universidade Estadual de Londrina, Londrina, 2010. Disponível em: <http://www.gaia.uel.br/media/uploads/gaia/TCC_Dorigan.pdf >. Acesso em: 26 abr. 2015.

FABRI, J. A.; BEGOSSO, L. R.; PESSÔA, M. S. P., SPÍNOLA, M. M. Desenvolvimento do Conceito sobre Fábrica de Software em Instituições de Ensino que possuem Cursos de Computação. Lavras – MG, Universidade Federal de Lavras, ProQuality - Qualidade na Produção de Software, Vol. 2, Núm. 1, Maio 2006.

FERNANDES, A. A.; TEIXEIRA, D. S. Fábrica de Software: Implantação e gestão de Operações, Atlas, São Paulo, 2004.

FERRARINI, J. E. A. Identificação e Valoração de Competências para o Desenvolvedor de Sistemas de Informação, na Visão dos Gestores de Fábrica de Software de Salvador. Salvador, 2006. Dissertação (Mestrado) – Universidade Federal da Bahia.

GAFFO, F. H.; BARROS, M. R.; BRANCHER, J. D. Aplicação da Proposta da ISO 3100 em ambientes de Desenvolvimento de Software, Universidade Estadual de Londrina, Londrina, 2012.

GERHARDT, T. E.; SILVEIRA, D. T. Métodos de Pesquisa, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

GODOY, A. S. Introdução à pesquisa qualitativa e suas possibilidades. Revista de Administração de Empresas, v. 35, n. 2, 1995.

GUEDES, R. Fatores que influenciam na migração dos MPS.BR para o CMMI nas empresas de software brasileiras. Pernambuco, 2015. Dissertação (Mestrado) – Universidade Federal de Pernambuco.

LEITE, L. M.; LUCRÉDIO, D. Desenvolvimento de Software utilizando o Framework Scrum: um Estudo de Caso. T.I.S. São Carlos, v. 3, n. 2, p114-121, 2014.

MARQUES, H. M.; SILVA, I. G. L.; RAMOS, R. T.; MACIEL, T. M. M. Fábricas de Software e o processo de desenvolvimento segundo a experiência da FábricaUm
MARTINS, J. C. C. Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML, 4 ed, Brasport, Rio de Janeiro, 2007.

MEDEIROS, V. N.; ANDRADE C. A. R.; ALMEIDA, E. S.; ALCUQUERQUE, J.; MEIRA S. Construindo uma fábrica de Software: da Concepção às Lições Aprendidas. Centro de Informática da Universidade Federal de Pernambuco. 2004. Disponível em: <<http://www.cin.ufpe.br/~in953/publications/papers>>. Acesso em: 13 mar.2015.

MELLO, J. Sofhar lança BI com PUC-PR. 2013. Disponível em: <<http://www.baguete.com.br/noticias/02/12/2013/sofhar-lanca-bi-com-puc-pr>>. Acessado em: 27 de jul.2015.

MOTTA, F. C. P. VASCONCELOS, I. G. Teoria Geral da Administração, São Paulo: Pioneira Thomson Learning, 2006.

MURPHY, P. A Review of Cleanroom Software Engineering. 2007.

NAISBITT, J. Megatrends: as dez grandes transformações ocorrendo na sociedade moderna. São Paulo: Nova Cultural, 1982.

NAZARENO, A.; BOENTE, P.; OLIVEIRA, F. S. G.; CARLOS, J.; ALVES, N. RUP como Metodologia de Desenvolvimento de Software para Obtenção da Qualidade de Software. 2005.

O'BRIEN, JAMES A. Sistemas de Informação e as Decisões Gerenciais na Era da Internet. São Paulo: Editora Saraiva, 2002.

OLIVEIRA, D. H.; NETO, A. C. Fábrica de Software: Promovendo a Criação de Empresas Competitivas em Tecnologia da Informação, 2003.

OLIVEIRA, S. A. de. A Evolução da Adoção do CMM nas Fábricas de Software Nacionais nos Últimos 10 Anos. Revista Computação Aplicada-UnG, v. 1, n. 1, p. 25-32, 2012.

OSIAS, C. S. FÁBRICA DE SOFTWARE: um estudo de caso na Dataprev, sob a ótica da estrutura organizacional. Rio de Janeiro, 2008. Dissertação (Mestrado) – Fundação Getúlio Vargas.

PACHECO, H. UFPA é destaque no desenvolvimento de Software na América Latina., 2008. Disponível em: < <http://www.portal.ufpa.br/imprensa/noticia.php?cod=2136>>. Acesso em: 13 mar.2015.

PEREIRA, P; TORREÃO, P; MARÇAL, A, S. Entendendo Scrum para Gerenciar Projetos de Forma Ágil. Mundo PM, Curitiba, v. 14, n. 3. Disponível em: <<http://www.siq.com.br/DOCS/EntendendoScrumparaGerenciarProjtosdeFormaAgil.pdf>>. Acesso em: 26 abr. 2015.

PESSOA FILHO, H. F. B. Um estudo analítico entre as abordagens de Engenharia de Requisitos nas Metodologias Ágeis XP, SCRUM e Crystal. Recife – PE, Universidade Federal de Pernambuco, 2011.

PESSOA. M. S. P. Processos e Projetos em uma Fábrica de Software eLab-TI. São Paulo, 2009. Escola Politécnica da Universidade de São Paulo.

PINTO. M. C. Aplicação de Arquitetura Pedagógica em Curso de Robótica Educacional com Hardware Livre. Rio de Janeiro, 2011. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro.

PMBOK, GUIA. Um guia do conjunto de conhecimentos em gerenciamento de projetos. In: Project Management Institute. 2004.

PRIKLADNICKI, R. Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas. In: Fórum de Educação em Engenharia de Software, 2.,2009, Fortaleza. Anais. Fortaleza: UFC, 2009.

QUINTELLA, H. L. M. M.; ALMEIDA, J. L. I. Fábrica de Software: análise do impacto na competitividade. Associação Educacional Dom Bosco. Seget, 2006.

QUIVY, R.; CAMPENHOUDT, L. V. Manual de Investigação em Ciências Sociais. 2a. ed. Lisboa: Gradiva, 1998.

REZENDE, D. A. Engenharia de software e sistemas de informação. Brasport, 2005.

ROCHA, T. A.; OLIVEIRA, S. R. B.; VASCONCELOS, A. M. L. Adequação de Processos para Fábricas de Software. VI Simpósio Internacional de Melhoria de Processos de Software, 2004. Disponível em:

<http://www.simpros.com.br/Apresentacoes_PDF/Artigos/Art_12_Simpros2004.pdf>.

Acessado em: 12 mar.2015.

SANTOS, A. C. C.; SETTE, J. P. F.; FILHO, A. T. A.; RAMOS, I. C.; SOUZA, L. S.; LIMA, L. A. L.; BACELAR, R. A.; CARVALHO, R. C. L.; SILVA, F. Q. B. Experiência Acadêmica de uma Fábrica de Software utilizando Scrum no Desenvolvimento de Software. Universidade Federal de Pernambuco. 2010.

SARINHO, V. Usando Atividades Práticas e Avaliação Contínua no Ensino de Engenharia de Software. In: XXV Congresso da Sociedade Brasileira de Computação-XIII WEI. Unisinos, São Leopoldo/RS, 2005.

SATO, D. T. Uso eficaz de métricas em métodos ágeis de desenvolvimento de software. São Paulo, 2007. Dissertação (Mestrado) – Instituto de Matemática e Estatística, Universidade de São Paulo.

SILVA, A. C. Sistema de Gerenciamento de Tarefas para usuários de Scrum. 115 p. Dissertação (Graduação em Engenharia Eletrônica e Computação) – Escola Politécnica – Departamento de Eletrônica e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

SILVERMAN, D. Interpreting Qualitative Data: methods for analysing talk, text and interaction. Reino Unido: SAGE Publications, 1993.

SOARES, F. S. F.; MARIZ, L. M. R. S.; CAVALCANTI, Y. C.; RODRIGUES, J. P.; BASTOS, P. R.; NETO, M. G., ALMEIDA, A. C. M.; PEREIRA, D. T. V.; ARAÚJO, T. S.; ALBUQUERQUE, R. S. M. C. J. Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software. Centro de Informática – Universidade Federal de Pernambuco, Recife - PE, 2007.

SOMMERVILLE, I. Agile software development. Software engineering (9th, pp. 56–81). Boston, Mass: Pearson, 2011.

TEIXEIRA, C. A. N.; CUKIERMAN, H. L. Apontamentos para Enriquecer o Perfil do Engenheiro de Software. Engenharia de Software, 2005.

TOLEDO, J. C. Fatores críticos de sucesso no gerenciamento de projetos de desenvolvimento de produto em empresas de base tecnológica de pequeno e médio porte. Gestão & Produção, v. 15, n. 1, 2008.

TOMOMITSU, C. K. A.; CAMARGO, V. L. S.; FILHO, A. N. Dimensões a considerar na análise dos problemas de ensino e aprendizagem de engenharia de software. Centro Paula Souza. 2006.

VASCO, C. G.; VITHOFT, M. H.; ESTANTE, P. R. C. Comparação entre Metodologias RUP e XP. Curitiba, PR. Pontifícia Universidade Católica do Paraná (PUCPR), 2005.

VERGARA, G. F. Implantação de Software como Serviço em uma Nuvem Privada para a Fábrica de Software da FGA. Brasília, Universidade de Brasília, 2014.

VERSIONONE; 9TH ANNUAL State of Agile™ Survey, 2015. Disponível em: <<http://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>>. Acessado em: 02 Ago.2015.

VIVACQUA, F. R. Fábricas de Software e a Academia: Análise da Formação Acadêmica em Informática no Município do Rio de Janeiro. Rio de Janeiro, 2009. Dissertação (Mestrado) - Fundação Getúlio Vargas.

ZANELLA, L. C. H. Metodologia de estudo e de pesquisa em administração. Florianópolis: Departamento de Ciências da Administração/UFSC, 2009.

APÊNDICE A – QUESTIONÁRIO APLICADO A GESTORES DE FÁBRICAS DE SOFTWARE EM OUTRAS INSTITUIÇÕES DE ENSINO SUPERIOR

O questionário enviado pode ser acessado por meio do seguinte endereço:
<https://docs.google.com/forms/d/1K-GNRYmXsWG4OJPelVX-qbGXxpCIL1aZlaeQMXmA4Qo/viewform>.

Abaixo encontra-se as perguntas em formato de imagem extraídas da página do formulário.



Questionário sobre Fábrica de Software em Instituições de Ensino Superior

Prezado(a) Senhor(a):


O Prof. Silas Dias Romanha, da Faculdade de Ciências Econômicas, Administrativas e da Computação Dom Bosco - FCEACDB -, aluno do curso de Mestrado Profissional em Engenharia de Produção da Universidade Estadual Paulista - UNESP-, campus Guaratinguetá, está conduzindo uma pesquisa sobre Fábricas de Software em instituições de ensino superior, que servirá de base para a dissertação de mestrado: "Adequação de um modelo de Fábrica de Software e implementação em uma instituição de ensino superior".

Para tanto, faz-se necessário coletar dados a respeito de experiências similares realizadas em instituições de ensino superior no Brasil. Por isso, gostaríamos de contar com sua colaboração, respondendo ao questionário a seguir, baseando suas respostas em sua percepção a respeito da natureza e dos resultados obtidos através da implementação do conceito de Fábrica de Software na instituição da qual faz parte.

Sua participação é muito importante e suas respostas serão tratadas de forma agregada, anônima e confidencial. Exclusivamente para os fins científicos desta dissertação.

O preenchimento do formulário deve levar de 5 a 10 minutos e as respostas podem ser enviadas até o dia 16/10.
Muito obrigado por sua participação!

Continuar »

 33% concluído

Questionário sobre Fábrica de Software em Instituições de Ensino Superior

*Obrigatório

Perfil do respondente

Nome: *

Qual a sua função na Fábrica de Software? *

- Analista de Requisitos
- Analista de Testes
- Arquiteto de Software
- Banco de Dados (DBA)
- Coordenador/Líder de equipe
- Diretor
- Documentador
- Gerente de Fábrica de Software
- Gerente de Projetos
- Programador/Implementador
- Projetista/Analista de Sistemas
- Web Designer
- Outro:

« Voltar

Continuar »



66% concluído

Em relação à Fábrica de Software de sua instituição, responda:

Quais foram os fatores que levaram à instituição a decidir pelo modelo de Fábrica de Software? *

Para você, quais foram as principais dificuldades encontradas durante o processo de implantação da Fábrica de Software? *

Como se dá a relação entre o conteúdo do(s) curso(s) de Tecnologia da Informação da instituição e a Fábrica de Software? *

A Fábrica é considerada parte do(s) curso(s) ou algo complementar e opcional para os alunos?

Quais metodologias de desenvolvimento de software são utilizadas no dia a dia da Fábrica? *

- CMM
- CMMI
- [MPS.BR](#)
- RUP
- XP (Extreme Programming)
- Cleanroom
- Família Crystal de Metodologias
- Scrum
- Outro:

Qual o impacto da substituição de membros da equipe de desenvolvedores no dia a dia das operações da Fábrica? Como lidam com essa situação? *

Historicamente, quais os principais motivos para a substituição dos membros das equipes de desenvolvimento da Fábrica? *

- Conclusão do curso (formatura)
- São contratados por outras empresas
- Não se adaptam à natureza do trabalho
- Outro:

Os projetos da Fábrica de Software são destinados a quais tipos de clientes? *

- Necessidades da própria instituição
- Pequenas empresas
- Grandes empresas
- Outras instituições de ensino
- Outro:

A Fábrica possui alguma certificação ou recebeu algum prêmio de reconhecimento? Qual? *

Considerando os três modelos de produção descritos abaixo, qual deles melhor representa o atual estágio da Fábrica de Software de sua instituição? *

- Desenvolvemos softwares inteiramente personalizados, elaborando projetos únicos adaptados especificamente às necessidades de cada cliente.
- Fabricamos produtos semipadronizados. Nossos softwares semi-prontos são mantidos em estoque e depois finalizados, adaptados e configurados de acordo com as necessidades específicas dos clientes no momento da instalação.
- Nossos software são similares a commodities, inteiramente padronizados, tendo componentes intercambiáveis. A produção em massa e os altos volumes são obtidos graças à utilização de procedimentos e formas de produção inteiramente padronizados.

Quais conselhos você daria para as instituições de ensino superior que planejam seguir o mesmo caminho e implementar uma Fábrica de Software acadêmica?

« Voltar

Enviar

100% concluído.

Nunca envie senhas pelo Formulários Google.

A tabela abaixo apresenta as respostas obtidas para a primeira metade das perguntas do formulário:

Nome:	Qual a sua função na Fábrica de Software?	Quais foram os fatores que levaram à instituição a decidir pelo modelo de Fábrica de Software?	Para você, quais foram as principais dificuldades encontradas durante o processo de implantação da Fábrica de Software?	Como se dá a relação entre o conteúdo do(s) curso(s) de Tecnologia da Informação da instituição e a Fábrica de Software?	Quais metodologias de desenvolvimento de software são utilizadas no dia a dia da Fábrica?
Carlos Augusto	Coordenador/Líder de equipe	A agilidade que o modelo proporciona, a escalabilidade e a adaptabilidade em relação à realidade da instituição.	Falta de experiência prática dos professores e alunos envolvidos. Curva de aprendizado.	Foram necessárias realizar mudanças nas ementas de algumas disciplinas do curso, como Gerência de Projetos e Arquitetura de Software, para que ensinassem as metodologias a serem utilizadas na Fábrica. A participação dos alunos vale parte da nota nestas disciplinas, de acordo com critério de cada professor.	Scrum
Jonas Damásio da Silva	Gerente de Projetos	A rica experiência prática que nossos alunos adquirem em métodos amplamente utilizados pelo mercado atual.	Rotatividade da equipe.	Considerada atividade complementar.	CMMI, Scrum
Augusto Ferrara	Gerente de Fábrica de Software	Queríamos que os alunos do nosso curso de Ciência da Computação tivessem maior contato com as melhores práticas de mercado, como o Scrum, melhorando assim a formação dos mesmos e também a avaliação do curso frente ao Ministério da Educação.	Lidar com equipes compostas por alunos que participavam de forma semi-presencial, devido à incompatibilidade de horários.	As disciplinas do curso ensinam as ferramentas, tecnologias e metodologias que são utilizadas pela fábrica. Alguns professores consideram a participação na fábrica como parte da avaliação de sua disciplina.	Adaptação do Scrum

A tabela abaixo apresenta as respostas obtidas para a segunda metade das perguntas do formulário:

Qual o impacto da substituição de membros da equipe de desenvolvedores no dia a dia das operações da Fábrica? Como lidam com essa situação?	Historicamente, quais os principais motivos para a substituição dos membros das equipes de desenvolvimento da Fábrica?	Os projetos da Fábrica de Software são destinados a quais tipos de clientes?	A Fábrica possui alguma certificação ou recebeu algum prêmio de reconhecimento? Qual?	Considerando os três modelos de produção descritos abaixo, qual deles melhor representa o atual estágio da Fábrica de Software de sua instituição?	Quais conselhos você daria para as instituições de ensino superior que planejam seguir o mesmo caminho e implementar uma Fábrica de Software acadêmica?
Este problema é minimizado na medida em que a base de conhecimento da Fábrica, assim como o repositório de artefatos cresce. No início causou atrasos e transtornos.	Conclusão do curso (formatura), São contratados por outras empresas	Pequenas empresas, Pessoas Físicas (Aplicativos)	Nível 2 do CMMI	Fabricamos produtos semipadronizados. Nossos softwares semi-prontos são mantidos em estoque e depois finalizados, adaptados e configurados de acordo com as necessidades específicas dos clientes no momento da instalação.	Comecem por projetos de baixa complexidade. Adaptem o Scrum à realidade da instituição e não o contrário. Busquem certificação assim que possível.
Uma grande dificuldade no início, mas minimizada na medida em que ganhamos maturidade em nossos processos.	Conclusão do curso (formatura), São contratados por outras empresas, Não se adaptam à natureza do trabalho	Necessidades da própria instituição	Ainda não.	Desenvolvemos softwares inteiramente personalizados, elaborando projetos únicos adaptados especificamente às necessidades de cada cliente.	Vale a pena, mas não espere retorno e satisfação imediatos.
Baixo, pois temos uma base de conhecimento sólida.	Conclusão do curso (formatura)	Necessidades da própria instituição, Pequenas empresas	Está nos planos para este ano iniciarmos a certificação em CMMI.	Desenvolvemos softwares inteiramente personalizados, elaborando projetos únicos adaptados especificamente às necessidades de cada cliente.	Realizem testes para selecionar os alunos participantes. Não economizem em infraestrutura e licenças necessárias.

APÊNDICE B – ESTUDOS DE CASO: FÁBRICAS DE SOFTWARE EM INSTITUIÇÕES DE ENSINO SUPERIOR NO BRASIL

Conforme metodologia de pesquisa definida para este trabalho, o Estudo de Caso, foi possível observar que no Brasil há diversos casos de sucesso da implementação do conceito de Fábricas de Software em instituições de ensino superior. A seguir serão apresentados alguns dos exemplos encontrados:

1. Universidade Federal do Pará

A UFPA é um dos bons exemplos que podemos encontrar no país. Sua Fábrica de Software já recebeu diversos prêmios nacionais e internacionais, como o 7º lugar dentre as 25 instituições de ensino de ciência e tecnologia da América Latina que atuam na área de Engenharia de Software em 2008, por meio de uma pesquisa realizada pela empresa alemã Bosh e o segundo lugar no Prêmio Dorgival Brandão Junior de Qualidade e Produtividade de Software do MCT pelo projeto do Software Livre WebAPSEE, em 2007.

Segundo a pesquisadora Carla Reis, uma das coordenadoras do projeto, evitar a fuga de cérebros é um dos principais objetivos do grupo. Ela afirma que esse tipo de iniciativa gera desenvolvimento e proporciona uma melhor qualidade de ensino, uma vez que os alunos se aprimoram na pesquisa científica e teórica e ainda aprendem a aplicar esse conhecimento, sempre que possível em parceria com empresas. (PACHECO, 2008).

2. Faculdade Lourenço Filho

Outro exemplo de sucesso é a Fábrica de Software da Faculdade Lourenço Filho, cujos responsáveis definem como um laboratório de prática onde a principal estratégia é o aprendizado a partir de vivências de desenvolvimento de software para resolução de problemas reais, executadas utilizando tecnologias de desenvolvimento de fábricas de software diferenciadas.

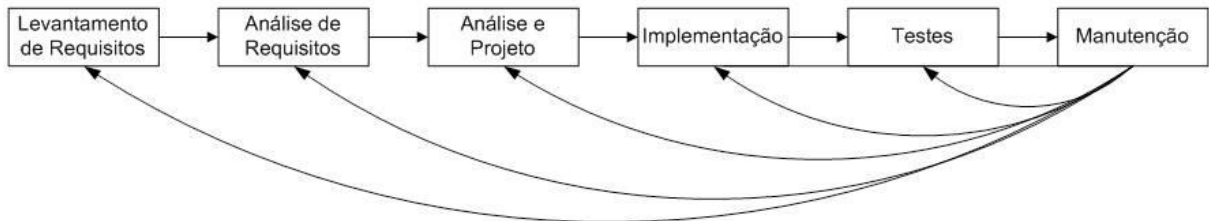
Figura 1: Logo da Fábrica de Software da Faculdade Lourenço Filho



Fonte: (<http://www.flf.edu.br/fabrica/home/quemsomos>).

Segundo eles, o papel da fábrica de software é desenvolver um produto (software), que é trabalhado por todos, em um laboratório de práticas que permeia todas as fases de sua construção, recorrendo, também, a aspectos teóricos, conforme exemplificado na Figura 2, retirada do site da Fábrica de Software da FLF:

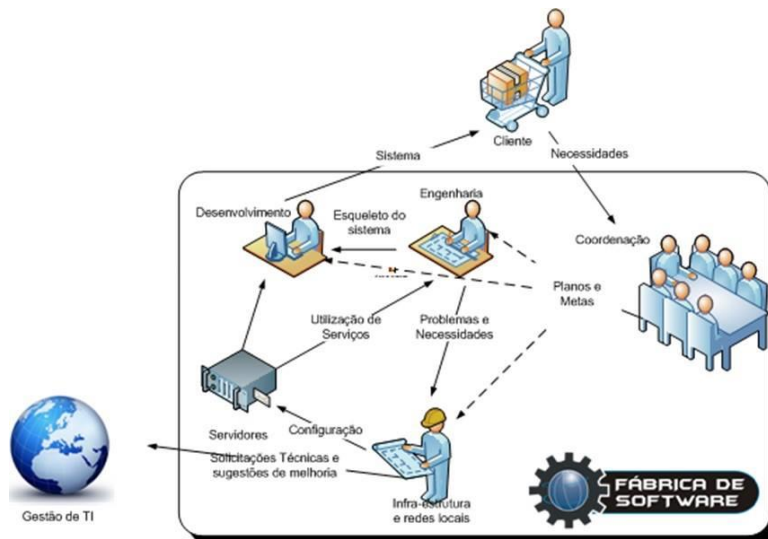
Figura 2: Etapas do desenvolvimento de software na Fábrica da Faculdade Lourenço Filho



Fonte: (<http://www.flf.edu.br/fabrica/home/quemsomos>).

Ainda de acordo com a Faculdade Lourenço Filho, para que estas etapas sejam realizadas, a equipe de trabalho é dividida em três áreas de estudo: Engenharia de Software, Linguagens de Programação e Ambientes e Redes, também conhecidos como os três pilares da Fábrica de Software, conforme mostrado na Figura 3:

Figura 3: Divisão da equipe e funcionamento da Fábrica de Software da FLF



Fonte: (<http://www.flf.edu.br/fabrica/home/quemsomos>).

Cada um dos pilares tem as seguintes funções: A primeira é responsável pelas três primeiras etapas de desenvolvimento de software (levantamento e análise de requisitos e elaboração do projeto). A segunda área de estudo é responsável pelas três últimas fases de

desenvolvimento (implementação, testes e manutenção). A última equipe é responsável por disponibilizar todo o ambiente e fornecer toda infraestrutura de apoio para ao desenvolvimento da solução proposta.

3. Faculdade de Tecnologia de Jundiaí

A Faculdade de Tecnologia de Jundiaí é uma das 13 faculdades do Centro Estadual de Educação Tecnológica Paula Souza do Governo do Estado de São Paulo. Além das faculdades, esta autarquia estadual é composta também por 102 escolas técnicas de ensino médio, distribuídas por 88 municípios do Estado de São Paulo, possuindo ao todo cerca de 10 mil alunos no ensino superior e 100 mil alunos no ensino técnico e de nível médio. Um dos cursos oferecidos pela FATEC Jundiaí é o de Graduação em Tecnologia em Informática com Ênfase em Gestão de Negócios. O curso além de operacionalizar os conceitos relacionados com a Tecnologia da Informação, possui também disciplinas voltadas para o desenvolvimento do espírito empreendedor dos alunos, mediante atividades que estimulam a criatividade e a inovação, além da formação em gestão de empresas.

A fábrica de software é considerada pela FATEC Jundiaí um importante elemento de desenvolvimento regional, pois acrescenta à graduação de seus alunos sólidas atividades de capacitação tecnológica, estimulando o empreendedorismo, que estimula o surgimento de projetos inovadores, que possam inclusive ser propostos, mediante a apresentação de plano de negócios, em incubadoras de software (OLIVEIRA; NETO, 2003)

4. Universidade Federal de São Carlos (UFSCar)

A Universidade Federal de São Carlos realizou um experimento de desenvolvimento de um software para uso interno fazendo uso da mão de obra de seu corpo acadêmico. O objetivo era criar uma solução sistêmica para o restaurante universitário, que gerenciava seus processos de maneira totalmente manual.

Para realizar a tarefa o grupo composto de 6 alunos passou por treinamentos específicos sobre a metodologia Scrum de desenvolvimento de software, apresentada no Tópico 3.3.5.

A faculdade registra que por se tratar de um ambiente acadêmico, fez-se necessário realizar algumas adaptações no modelo de *sprints* proposto pela metodologia Scrum, no que tange ao período de realização de cada *sprint* comumente fixados entre duas e quatro semanas, que passou a variar entre quarenta e um e oitenta e quatro dias. A rotatividade do Scrum Master

foi outra alteração realizada, pensada para que cada membro da equipe praticasse as habilidades requeridas. Os seis alunos que compunham a equipe inicial possuíam pouco conhecimento em relação ao framework e às tecnologias que seriam utilizadas para o desenvolvimento. Para minimizar o problema da inexperiência, foram ministradas disciplinas que compreendiam o conhecimento necessário a respeito das técnicas a serem utilizadas.

O Scrum prevê a realização de reuniões presenciais diárias, afim de que os desenvolvedores relatem o andamento de suas atividades e se situem em relação ao projeto como um todo, porém realizar reuniões presenciais com esta frequência torna-se uma atividade complexa quando se tratando de uma equipe distribuída. Para minimizar este problema, foi utilizada uma ferramenta on-line de comunicação instantânea, e as discussões passaram a ocorrer inicialmente uma vez por semana. Foi utilizada também uma ferramenta web para realizar o acompanhamento periódico do trabalho, o Product Backlog. As Sprint Backlog e os gráficos Burndown foram compartilhados por todos os membros a partir da utilização de um instrumento de produção e armazenamento de arquivos na “nuvem”, mantendo assim a eficiência na visibilidade e interação.

No que se refere aos aspectos técnicos, foi utilizada uma plataforma computacional baseada em software livre e as implementações utilizaram as linguagens Java, CSS e HTML, por meio do ambiente de desenvolvimento NetBeans 7.2. Foi utilizado o servidor Glassfish 3.2.1 para execução do sistema e o sistema gerenciador de banco de dados (SGDB) denominado PostgreSQL, versão 9.2. Para facilitar o reaproveitamento de código, foi utilizado o padrão MVC, que separa a representação gráfica das informações em camadas.

Coube ao Product Owner, no caso, o chefe do restaurante, explicar o funcionamento do mesmo, apontando inclusive aqueles processos que deveriam ter prioridade de desenvolvimento. Para facilitar futuras consultas, a equipe de desenvolvedores gravou em áudio a reunião. Durante a avaliação, foram realizadas as primeiras reuniões por meio de um software para vídeo chamadas, destinadas à definição do planejamento do projeto. Após concluírem que haviam atingido um nível satisfatório de conhecimento do sistema, o grupo estabeleceu as primeiras tarefas do Product Backlog. Em seguida, dois professores iniciaram um trabalho de auxílio aos alunos, atuando como base para a reafirmação dos valores do framework e também como elo na comunicação entre o time e o dono do produto, dada a maior experiência dos professores no que diz respeito a entender necessidades de negócios.

A principal contribuição deste trabalho foi, portanto, ter identificado os empecilhos e possíveis soluções ao implantar o Scrum em um ambiente acadêmico, levando em consideração

a inexperiência dos membros das equipes no que diz respeito à esta abordagem. (LEITE e LUCRÉDIO, 2014).

5. Universidade Federal de Pernambuco (UFPE)

Durante as pesquisas realizadas, foi encontrado um estudo de caso de um trabalho realizado por uma fábrica de desenvolvimento de software de código aberto formada por dez alunos do curso de mestrado do Centro de Informática da Universidade Federal de Pernambuco. A estrutura da fábrica possui um Comitê Gestor, que fica responsável pela tomada de decisões estratégicas e por cinco Unidades de Produção, cada uma com atribuições bem definidas e complementares.

Figura 4: Estrutura Organizacional da Fábrica de Software da UFPE



Fonte: (SOARES, MARIZ, CAVALCANTI, RODRIGUES, BASTOS, NETO, ALMEIDA, PEREIRA, ARAÚJO e ALBUQUERQUE, 2007)

O processo adotado pela fábrica baseia-se nas melhores práticas de engenharia de software, no RUP e no Manifesto Ágil (AGILE MANIFESTO, 2001). Trata-se de um processo caracterizado como um processo social, diferentemente dos processos tradicionais, pois é baseado em esforço colaborativo e em gerência comunitária, além de ser executado de forma distribuída, assíncrona e descontínua.

Durante o trabalho analisado, foi possível observar diversos aspectos da metodologia Scrum de desenvolvimento de software. Primeiramente, foi realizado pelo comitê da fábrica de

software um estudo de viabilidade baseando-se na visão apresentada pelo cliente, neste caso, o product owner. Em seguida, todo o product backlog foi descrito em uma proposta comercial para quem então este product backlog fosse devidamente priorizado e que dessa priorização fossem criadas as sprints do projeto. Cada sprint tinha duração de quinze dias e ao final de cada uma, era disponibilizado um conjunto de novo produto

Os itens do backlog com maior valor de negócio eram apontados pelo product owner durante reuniões assíncronas realizadas em cada sprint. Em seguida, a complexidade desses itens era avaliada pela equipe por meio de fóruns e listas de discussões, de modo a determinar o que iria ser feito na Sprint, levando em consideração sua capacidade de produção.

Em seguida era feito um detalhamento dos requisitos para que então o time selecionasse os itens de backlog priorizados, dividindo-os em tarefas de até 1 semana por participante. Para fazer o devido planejamento e acompanhamento do projeto, a equipe fez uso da ferramenta XPlanner, especializada em projetos que utilizam métodos ágeis de desenvolvimento.

Durante cada sprint, a equipe simulava o Daily Scrum Meeting por meio de fóruns e listas de discussões, possibilitando que, diariamente, todos do time pudessem acompanhar o andamento das atividades e as dificuldades encontrados até aquele momento. Um dos integrantes do comitê da fábrica tinha a responsabilidade de exercer o papel do Scrum Master, visando resolver os impedimentos reportados por qualquer membro do time.

Tanto a comunicação entre os membros da equipe como a comunicação entre a equipe e seu product owner, seja para aprovação de documentos, seja para o esclarecimento de dúvidas ou alinhamento do andamento da sprint, eram realizadas por meio de fóruns, e-mails ou ferramentas de comunicação assíncrona. No final de cada Sprint era realizada uma reunião síncrona, por meio de ferramentas como o Skype, com o objetivo de apresentar ao cliente os resultados obtidos até aquele momento. Em seguida, ocorria uma retrospectiva do Sprint, da qual todos os integrantes do time eram convidados a participar, sendo uma reunião presencial na maioria das vezes, onde eram documentadas por um integrante da equipe as lições aprendidas, assim como os pontos fortes, fracos e sugestões de melhoria de cada participante.

Ao longo de cada Sprint eram coletadas e analisadas algumas métricas com o objetivo de otimizar o uso do processo adaptado. Algumas delas tinham como objetivo verificar a eficácia do processo de estimativas de esforço adotado. Em um projeto específico, tomado como exemplo, ficou constatado que a variação média do esforço realizado pela equipe por sprint ficou em torno de 15% do valor estimado. Enquanto que o fator produtividade sofreu uma melhora significativa: indo de 18 homem-hora para cada ponto de caso de uso no início do projeto e estabilizando em 10 homem-hora para cada ponto de caso de uso produzido.

Concluiu-se, portanto, que nem todas as práticas do Scrum eram diretamente aplicadas ao contexto de desenvolvimento distribuído de software, como por exemplo, as reuniões diárias de 15 minutos sugeridas pelo Scrum, que foram substituída pela comunicação assíncrona, dentre outros aspectos. Foi possível fazerem uso de diversos aspectos de desenvolvimento ágil sem que fossem comprometidas as particularidades exigidas por esses tipos de projetos, mesmo o Scrum não cobrindo todas as características específicas para equipes geograficamente distribuídas. (SOARES, et al, 2007).

APÊNDICE C – METODOLOGIAS E BOAS PRÁTICAS DE DESENVOLVIMENTO DE SOFTWARE

Nesta seção, serão apresentados diferentes modelos utilizados no processo de desenvolvimento de software. Ao final deste estudo, serão apresentados aqueles que melhor atendem às necessidades de uma Fábrica de Software em ambiente acadêmico de ensino superior.

1. CMM e CMMI

CMM e CMMI são modelos de gestão de processo de software desenvolvidos pelo SEI - Software Engineering Institute, que é mantido com verbas do departamento de defesa dos Estados Unidos da América e é gerenciado pela Universidade Carnegie Mellon, em Pittsburgh. Também conhecido como SW-CMM, O CMM surgiu da necessidade do departamento de defesa Norte-Americano em determinar a capacidade de seus fornecedores de desenvolver um software atendendo de maneira satisfatória as especificações do projeto.

O CMM caracteriza-se por ser composto de cinco níveis de maturidade para o processo de desenvolvimento e manutenção de software (FERNANDES e TEIXEIRA, 2004), são eles:

No primeiro nível, a gestão do processo não é aplicada, considerado inclusive uma situação caótica, pois o sucesso do projeto é entregue exclusivamente para as pessoas.

No segundo nível são estabelecidas políticas e procedimentos para a gestão do projeto de software, com a implantação de controles básicos, e exige: obrigação de estabelecer e seguir padrões para projetos de software; comprometimento com as estimativas obtidas por meio da interpretação dos requisitos do projeto; monitoramento de custos, cronograma e funcionalidades do projeto; políticas e procedimentos para a contratação de subcontratados; gerenciamento e controle das mudanças, com rastreabilidade do software produzido.

No terceiro nível é documentado todo o processo padrão de desenvolvimento e manutenção de software. Cada processo pode ser adaptado a partir do padrão, para poder atender às suas particularidades. Há também a busca pela melhoria contínua do processo de software. Deve ser estabelecido um grupo responsável pelo processo de software na organização, que deve promover o planejamento e implementação de amplo programa de treinamento, além de coordenar a comunicação entre os grupos de pessoas envolvidos no processo.

No quarto nível, a organização estabelece metas quantitativas de qualidade para o processo de software e para o produto originado, sendo necessário: mensurar qualidade e produtividade; implementar uma base de dados com controle individual de cada projeto; utilizar as medições para realizar melhorias no processo de software; conhecer e gerenciar os riscos do projeto.

Já no quinto e último nível o foco é na melhoria contínua, obrigando a organização a atuar de forma fortemente preventiva, planejar e gerenciar mudanças no processo e nas tecnologias utilizadas.

Por sua vez, o CMMI é do que uma evolução do CMM, pois além de possuir os cinco estágios de maturidade e de promover a melhoria dos processos de desenvolvimento de produtos e serviços, ele também controla sua aquisição e manutenção, adicionando ao CMM disciplinas de gestão de fornecimento, de processos e de projeto (GUEDES, RHAVY, 2014).

Chrissis, Konrad e Shrum (2003) apresentam o CMMI como um novo modelo de melhoria de processos. Para os autores, não há dúvidas quanto as vantagens do CMMI sobre os outros modelos de processos de software. Entre as vantagens eles citam, por exemplo:

- oferece uma cobertura mais detalhada do ciclo de vida do produto;
- incorpora as várias lições aprendidas durante o desenvolvimento e a manutenção dos modelos que o inspiraram e precederam, incluindo a solução para vários problemas que eram encontrados no CMM;
- as organizações pioneiras, ao atingirem os níveis quatro e cinco do CMM, relataram suas dificuldades e sucessos, que foram utilizadas para melhorar as práticas de níveis mais altos do CMMI;
- provê uma oportunidade de eliminar obstáculos e barreiras que normalmente existem em diferentes partes da organização, e que geralmente não são tratados por outros modelos de melhoria de processo;
- promove uma colaboração entre engenharia de software e engenharia de sistemas, mudando o foco para o produto final e para os processos associados a ele. Além disso, permite que o treinamento no modelo e nas avaliações seja simplificado;
- é valioso para organizações que produzem o software com atividade fim. As funções de engenharia de sistemas, normalmente não tratadas detalhadamente por outros modelos voltados apenas para o software, são extremamente valiosas;
- permite que o usuário selecione o modelo de representação mais adequada para o negócio;

- apesar do foco inicial do CMM ser na engenharia de produtos e serviços, foi definido para atender a outras áreas também, como o suporte a um processo de melhoria corporativo.

2. MPS.BR

No Brasil, no que se refere à qualidade no processo de software, há dois grupos de empresas (SOFTEX, 2005a): o primeiro é composto pelas poucas empresas de grande porte com potencial exportador de software e que desejam atingir elevada maturidade nos processos de qualidade de software, equivalente aos níveis 4 e 5 do CMMI; já o segundo grupo é formado pela grande maioria das empresas, ou seja, as micro, pequenas e médias, que dispõe de poucos recursos e que desejam melhorar seus processos de software, mantendo uma compatibilidade com os modelos de melhoria de software já existentes, equivalendo aos níveis 2 e 3 do CMMI.

Pensando no segundo grupo de empresas, tendo em vista que o processo de implantação do CMMI é bastante demorado, podendo levar até 10 anos para certificar empresas de maior porte nos níveis 4 ou 5 (SOFTEX, 2005a, p. 5), foi criado o modelo MPS.BR, que tem como propósito principal aumentar a capacitação das empresas brasileiras para o desenvolvimento de software de qualidade. O modelo permite que estas empresas implantem princípios de engenharia de software em consonância com as principais abordagens internacionais, por meio de um processo adequado à realidade das empresas brasileiras.

O MPS.BR foi desenvolvido pelo sistema SOFTEX, por Instituições de Ensino, Pesquisa e Centros Tecnológicos (COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação da Universidade Federal do Rio de Janeiro, CESAR - Centro de Estudos e Sistemas Avançados de Recife e CenPRA - Centro de Pesquisas Renato Archer) e também por uma sociedade de economia mista, a CELEPAR - Companhia de Informática do Paraná.

Para manter a compatibilidade, o MPS.BR tem sua base. Assim como nos outros modelos, este também é dividido em níveis, conforme a seguir:

- Nível G - Parcialmente Gerenciado
 - Gerência de Projetos - GPR
 - Gerência de Requisitos - GRE
- Nível F - Gerenciado
 - Gerência de Configuração - GCO
 - Garantia da Qualidade - GQA
 - Medição - MED
 - Aquisição - AQU

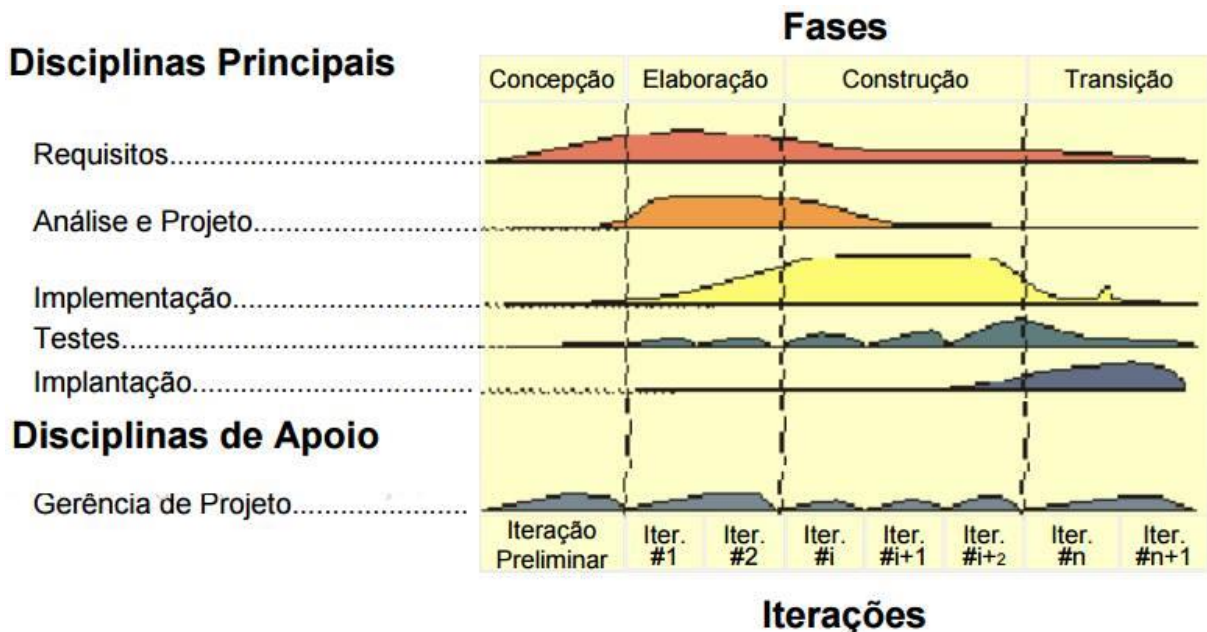
- Nível E - Parcialmente Definido
 - Treinamento - TRE
 - Definição do Processo Organizacional - DFP
 - Avaliação e Melhoria do Processo Organizacional - AMP
 - Adaptação do Processo para Gerência de Projeto - APG
- Nível D - Largamente Definido
 - Desenvolvimento de Requisitos - DRE
 - Solução Técnica - STE
 - Validação - VAL
 - Verificação - VER
 - Integração do Produto - ITP
 - Instalação do Produto - ISP
 - Liberação do Produto - LIP
- Nível C - Definido
 - Gerência de Riscos - GRI
 - Análise de Decisão e Resolução - ADR
- Nível B - Gerenciado Quantitativamente
 - Desempenho do Processo Organizacional - DEP
 - Gerência Quantitativa do Projeto - GQP
- Nível A - Em Otimização
 - Inovação e Implantação na Organização - IIO
 - Análise e Resolução de Causas – ARC

3. RUP

Um dos processos mais adotados é o RUP (Rational Unified Process). Nele, os custos e prazos são definidos no início do projeto. Este processo oferece um framework centralizado e é baseado em boas práticas de desenvolvimento.

Empregado principalmente nas fábricas de projeto e outsourcing, no RUP os processos são classificados como tradicionais, possuindo uma previsibilidade maior dos requisitos do sistema. Este tipo de processo considera a especificação de requisitos uma etapa importante, a qual consiste em um planejamento detalhado com fases sequenciais e na geração de artefatos para a fase seguinte (MARTINS,2007).

Figura 1: Arquitetura Geral do RUP



Fonte: (MARTINS, 2007).

4. Extreme Programming (XP)

Programação Extrema (do inglês eXtreme Programming), ou simplesmente XP, é uma metodologia ágil para equipes pequenas e médias e que irão desenvolver software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software. (SATO, TOSHIKI, 2007)

No XP, as iterações tendem a serem curtas, o que gera frequentemente novas versões do produto para o cliente, também chamadas de *releases*. Ao final de cada iteração, todos os comentários, opiniões e lições aprendidas são documentados e servem de base de conhecimento para a próxima iteração. O principal objetivo do XP é tornar o projeto mais flexível, diminuindo o custo e o impacto de possíveis mudanças.

O ciclo de vida de um projeto que utiliza a metodologia XP é dividido em seis fases distintas:

1. Exploração: fase em que o cliente descreve as funcionalidades desejadas para o primeiro release no formato de cartões, sendo um por funcionalidade.
2. Planejamento: é quando ocorre, juntamente com o cliente, a definição das prioridades dentre as funcionalidades previamente previstas. É nesta etapa que os desenvolvedores estimam o esforço e o cronograma para cada uma das funcionalidades.

3. Iterações para Release: aqui ocorrem quantas iterações forem necessárias até que o primeiro release seja completado. O objetivo da primeira iteração é criar a arquitetura do sistema, já nas iterações seguintes são adicionadas às funcionalidades seguindo as prioridades previamente estabelecidas.

4. Productionizing (Validação para Produção): são feitas verificações e testes extensivos para validar o software para seja utilizado em ambientes de produção.

5. Manutenção: uma vez lançado o primeiro release do sistema para produção, são geradas novas versões do sistema com novas funcionalidades, sempre que necessário.

6. Morte: quando não há mais funcionalidades a serem implementadas e o cliente está satisfeito com produto que tem em mãos.

5. Cleanroom

A metodologia denominada de Cleanroom é considerada trabalhosa pelos padrões da Engenharia de Software, mas muito difundida entre grandes projetos corporativos.

Faz parte do processo uma análise apurada das funções por meio do método de revisão-par, que tem por objetivo verificar se tais funções realmente fazem o que foram especificadas a fazer. Por analogia, pode-se comparar esta metodologia com as salas limpas (“clear rooms”, em inglês) na fabricação de semicondutores, que eliminam a necessidade de se limpar wafers de silício pelo fato de que eles nunca começam sujos. O desenvolvimento Cleanroom remove a necessidade de depuração do programa, assegurando que os erros nunca começam introduzidos no sistema. Esta abordagem baseia-se nas seguintes estratégias (MURPHY, 2007):

- Especificação Formal: o software a ser desenvolvido é baseado em especificado formais.
- Desenvolvimento Incremental: o software é dividido em incrementos, que são desenvolvidos e validados separadamente.
- Programação Estruturada: o processo de codificação de um programa é definido como um processo de refinamentos sucessivos da especificação.
- Verificação Estática: o software é verificado estaticamente por meio de inspeções rigorosas.
- Testes Estatísticos de Sistema: cada incremento de software é testado estatisticamente para determinar sua confiabilidade. Esses testes são baseados em um perfil operacional desenvolvido em paralelo com a especificação do sistema.

6. Família Crystal de Metodologia (Crystal Family of Methodologies)

Tratam-se de uma família de métodos ágeis de desenvolvimento de software. Os membros da família Crystal utilizam uma identificação visual para indicar intensidade, de modo que quanto mais escura a cor, maior é a complexidade do projeto (PESSOA, 2011).

Todos os métodos membros da família Crystal compartilham algumas características em comum, tais como:

- O desenvolvimento incremental com ciclos de no máximo quatro meses;
- Maior ênfase na comunicação e cooperação entre as pessoas;
- A não limitação de quaisquer práticas de desenvolvimento, ferramentas ou produtos de trabalho;
- Incorporação de objetivos para reduzir produtos de trabalho intermediários e desenvolvê-los como projetos evoluídos.

O ciclo de vida desta família de metodologia é baseado 10 práticas principais, são elas:

- Staging: Nome da etapa em que é feito o planejamento do próximo incremento do sistema, onde a equipe seleciona os requisitos que serão implementados na iteração e o prazo para sua entrega;
- Edição e revisão: Consiste na construção, demonstração e revisão dos objetivos do incremento;
- Monitoramento: Onde o processo é monitorado com relação ao seu progresso e também quanto à estabilidade da equipe;
- Paralelismo e fluxo: Característica que permite que as diferentes equipes possam operar com máximo paralelismo possível. Isto é garantido acompanhando e medindo a estabilidade e o sincronismo entre as equipes;
- Inspeções de usuários: A cada incremento são sugeridas que sejam feitas de duas a três inspeções por usuários a cada incremento;
- Workshops refletivos: são reuniões que ocorrem antes e depois de cada iteração com objetivo de analisar o progresso do projeto;
- Local matters: Procedimentos a serem aplicados e que podem variar de acordo com o tipo de projeto;
- Produtos de Trabalho (Work Products): composto pela sequência de lançamento, os modelos de objetos comuns, o manual do usuário, os casos de teste e migração de código; casos de uso e descrições de funcionalidade; documento de requisitos.

- Padrões (Standards): padronização de notações, convenções de produto e qualidade usadas no projeto.
- Ferramentas (Tools): ferramentas mínimas a serem utilizadas, como compiladores, gerenciadores de versão e configuração, ferramentas de versionamento, de programação, de teste, de comunicação, de monitoramento de projeto, de desenho e de aferimento de performance.

7. Scrum

Segundo Takeuchi e Nonaka (1986), o Scrum é um método ágil que foi concebido como um estilo de gerenciamento de projetos em empresas de fabricação de automóveis e produtos de consumo. O termo é baseado em um comportamento dos atletas que praticam o esporte coletivo Rugby e foi escolhido pois os atletas deste esporte e os desenvolvedores, em ambos os casos agem como uma unidade integrada, onde cada membro da equipe desempenha um papel específico e todos buscam um objetivo comum. Takeuchi e Nonaka também afirmam em seu artigo que projetos com equipes pequenas e multidisciplinares produziram melhores resultados, fato que os autores associaram à formação Scrum (CARVALHO e MELLO, 2012;).

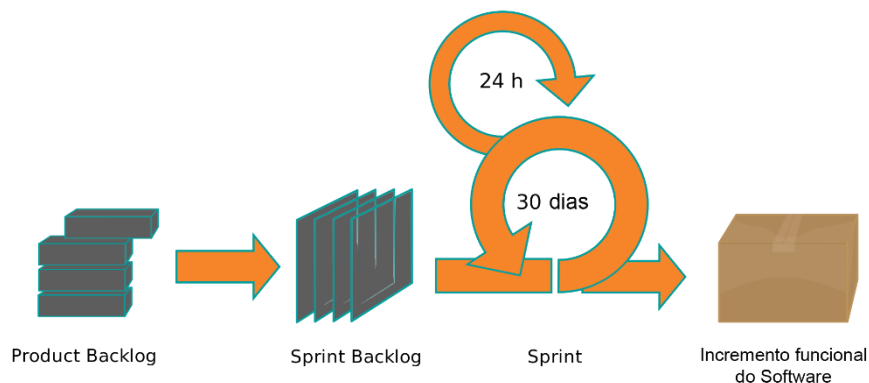
O papel do Scrum, segundo Schwaber e Sutherland (2013), é melhorar as práticas de desenvolvimento a partir da geração de evidências de sua eficácia, fornecendo um framework dentro do qual produtos de maior complexidade podem vir a ser desenvolvidos. Estes autores ainda afirmam que devido ao fato de ser fundamentado na teoria de controle de processos empíricos, o Scrum visa aperfeiçoar a previsibilidade e controlar riscos através de uma abordagem iterativa e incremental.

Segundo Silva, (2012), no Scrum, o desenvolvimento de software é feito da seguinte maneira:

- 1º. O representante do cliente, chamado de Product Owner, gera um documento de nome Product Backlog, onde estarão definidas as necessidades do mesmo.
- 2º. Coloca-se os itens desta relação em ordem de prioridade.
- 3º. Em seguida, a lista é apresentada à equipe de desenvolvimento, chamada de Team, em inglês, ou “Time”, que faz uma estimativa do tempo necessário para atender aos requisitos presentes na mesma.
- 4º. Realiza-se então uma Reunião de Planejamento da Sprint (Scrum Planning Meeting) com o objetivo de estabelecer quais funcionalidades serão implementadas em um período predeterminado, períodos chamados de Sprint.

- 5°. A fim de possibilitar o acompanhamento do andamento do trabalho, os integrantes do grupo informam diariamente o estado de suas atividades em reuniões rápidas chamadas de Daily Meeting, viabilizadas pela figura do Scrum Master, que é o responsável por ajudar o time a adotar o framework e eliminar possíveis impedimentos.
- 6°. Dois resultados distintos ajudam a estimar os esforços restantes ao longo do tempo: o primeiro abrangendo todo o projeto, chamado de Release Burndown e o segundo abrangendo apenas a iteração (Sprint Backlog).
- 7°. Após cada ciclo é realizada a revisão da Sprint (Sprint Review), onde a nova versão do sistema é apresentada, avaliada e as próximas tarefas são definidas, assim como uma retrospectiva da Sprint (Sprint Retrospective), onde são revisadas as contribuições buscadas soluções para os problemas encontrados.

Figura 2: Exemplo de funcionamento do Scrum



Fonte: (DORIGAN, 2010)

Pesquisas recentes apontam que 56% das empresas que utilizam metodologias ágeis, adotam o Scrum em seu dia a dia (VERSIONONE, 2015).

8°.ISO

A área de desenvolvimento de software também é regulamentada por meio da ISO. Para isso, foi criada a ISO 9000-3, que é uma interpretação da ISO 9000 voltada para o desenvolvimento e fornecimento de software (FERNANDES e TEIXEIRA, 2004).

8.1. ISO/IEC 12207

Foi publicada em 1995 como norma internacional, sendo a versão brasileira (incluindo as iniciais NBR) publicada em 1998. Em 2002 e 2004 passou por melhorias conhecidas como emendas 1 e 2.

A norma ISO/IEC 12207 e suas emendas 1 e 2 estabelecem uma arquitetura comum para o ciclo de vida de processos de software com uma terminologia bem definida. Esta norma categoriza os processos de três formas (FERNANDES e TEIXEIRA, 2004):

- Primários: fornecimento, desenvolvimento, operação e manutenção;
- De suporte: documentação, gestão de configuração, garantia da qualidade, verificação, validação, revisões conjuntas, auditorias e resolução de problemas;
- Organizacionais: gestão, infraestrutura, melhoria e treinamento.

Graças à compatibilidade, os processos definidos na NBR ISO/IEC 12207 podem ser utilizados como referência na implantação e avaliação do MR-MPS. Durante o processo de adaptação, é possível modificar, retirar ou incluir processos que não sejam pertinentes ao negócio (SOFTEX, 2005a).

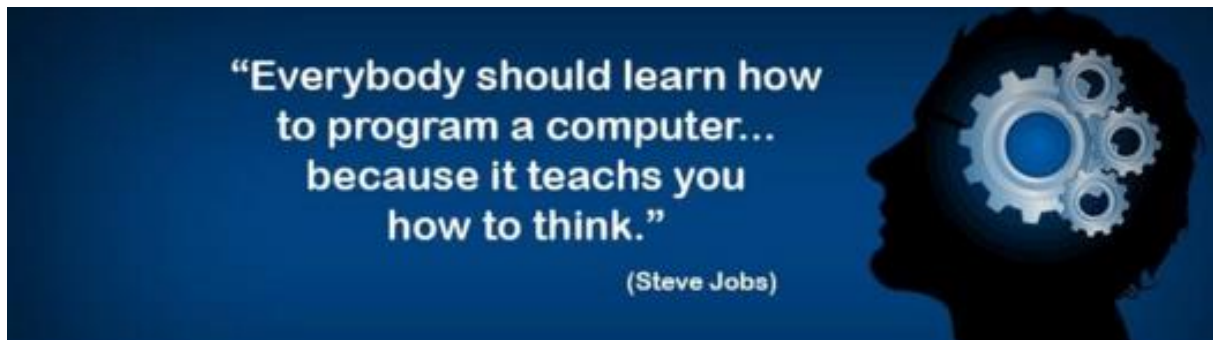
8.2. ISO/IEC 15504

Motivada pelas necessidades do governo britânico de definir um processo de avaliação de software, a ISO utilizou como referência um estudo de 1992, que indicava a importância da criação de um padrão para a melhoria de processos de software e a determinação de sua capacidade (FERNANDES e TEIXEIRA, 2004).

O projeto SPYCE (Software Process Improvement and Capability Determination), foi construído com o cuidado de englobar os métodos e normas já existentes na época (os principais eram o CMM e a ISO 9001). Assim, a ISO/IEC 15504 permite a realização de avaliações de processos de software com dois objetivos (SOFTEX, 2005a):

- A melhoria de processos, obtida por meio da auto avaliação que gera um perfil de processos, identificando os pontos fortes e fracos da organização;
- A determinação da capacidade de processos de uma unidade organizacional por meio da avaliação de um fornecedor em potencial, obtendo o perfil de capacidade que viabiliza a decisão de contratação do mesmo.

**APÊNDICE D – FORMULÁRIO DE INSCRIÇÃO DE ALUNOS PARA
TRABALHO REMOTO VOLUNTÁRIO NA FÁBRICA DE SOFTWARE DA AEDB**



Solicitação de participação na Fábrica de Software da AEDB

- Ao preencher este formulário, você concorda em participar do projeto sem qualquer tipo de remuneração ou vínculo empregatício.
- Trata-se de um trabalho voluntário que visa a obtenção de conhecimento prático em linguagens de programação e gerência de projetos de software.
- A participação nas atividades da Fábrica poderá valer horas complementares para os alunos participantes.

Pré requisitos para participar das atividades:

- Ter disponibilidade para comparecer à faculdade aos sábados de manhã e a tarde eventualmente.
- Ter concluído os seguintes cursos online:

Lógica de Programação:

[http://www.softblue.com.br/site/curso/id/6/CURSO+LOGICA DE PROGRAMACAO BASICO ON LINE L006](http://www.softblue.com.br/site/curso/id/6/CURSO+LOGICA+DE+PROGRAMACAO+BASICO+ON+LINE+L006)

Conceitos básicos do desenvolvimento de software:

http://www.microsoftvirtualacademy.com/training-courses/software-development-fundamentals_ptb

Fundamentos do C# com Clean Code: <http://www.microsoftvirtualacademy.com/training-courses/fundamentos-do-c-com-clean-code>

SQL Completo:

[http://www.softblue.com.br/site/curso/id/3/CURSO+SQL COMPLETO BASICO AO AVANCADO ON LI NE BD03](http://www.softblue.com.br/site/curso/id/3/CURSO+SQL+COMPLETO+BASICO+AO+AVANCADO+ON+LINE+BD03)

- Os certificados dos 4 cursos acima devem ter sido enviados para o e-mail "professor.silasromanha@gmail.com"

Continuar »



33% concluído

Solicitação de participação na Fábrica de Software da AEDB

*Obrigatório

Dados pessoais

Nome completo: *

Matricula

Email principal: *

Celular: *

Descreva o que te atrai na área de desenvolvimento de softwares: *

« Voltar

Continuar »



66% concluído

Solicitação de participação na Fábrica de Software da AEDB

*Obrigatório

Experiência

Possuo alguma experiência extra classe com: *

- C#.NET
- JAVA
- VB.NET
- ASP.NET
- PHP
- Desenvolvimento mobile
- Desenvolvimento de sites
- Desenvolvimento desktop
- Não possui qualquer experiência extra classe
- Outros:

Existe algum aplicativo Windows Desktop desenvolvido por você que esteja atualmente em uso em algum lugar? *

- Sim
- Não

Existe algum aplicativo Mobile desenvolvido por você que esteja atualmente em uso em algum lugar? *

- Sim
- Não

Existe algum aplicativo Web desenvolvido por você que esteja atualmente em uso em algum lugar? *

- Sim
- Não

Existe algum Website desenvolvido por você que esteja atualmente em uso em algum lugar? *

- Sim
- Não

Você atualmente está trabalhando/estagiando na área de desenvolvimento de software? *

- Sim
- Não

Qual o seu nível de entendimento de textos e vídeos no idioma Inglês? *

- Básico
- Intermediário
- Avançado

[← Voltar](#)

[Enviar](#)

100% concluído.

Nunca envie senhas pelo Formulário Google.

APÊNDICE E – PESQUISA DE SATISFAÇÃO DOS PRODUTOS GERADOS PELA FÁBRICA DE SOFTWARE DA AEDB

O questionário enviado pode ser acessado por meio do seguinte endereço: https://docs.google.com/forms/d/1m4i_m0UIESO6PyM5n3oRrxnlfpkb0Ehastwyueu0ugQ/vie_wform. Abaixo encontra-se as perguntas em formato de imagem extraídas da página do formulário:

Pesquisa de satisfação da Fábrica de Software da AEDB

*Obrigatório

Nome completo: *

Sua resposta

Empresa/Departamento: *

Sua resposta

Cargo que ocupa:

Sua resposta

E-mail para contato: *

Sua resposta

Sobre qual produto deseja opinar? *

Escolher

Em uma escala de 1 a 5, qual o seu nível de satisfação geral com a solução sistêmica selecionada? *

1 2 3 4 5

Muito insatisfeito Muito satisfeito

Escreva aqui alguma sugestão ou crítica, relacionada ao produto em questão, se desejar.

Sua resposta

ENVIAR

APÊNDICE F – TABULAÇÃO DA REVISÃO LITERÁRIA

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
GAFFO, BARROS, BRANCHER (2012)	APLICAÇÃO DA PROPOSTA DA ISO 31000 EM AMBIENTES DE DESENVOLVIMENTO DE SOFTWARE	DETALHAR APLICAÇÃO DA ISO 31000, EM CONJUNTO COM AS PRÁTICAS DO PMBOK, AO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE DA GAIA QUE É UMA FÁBRICA DE SOFTWARE MANTIDA PELO DEPARTAMENTO DE COMPUTAÇÃO DA UNIVERSIDADE ESTADUAL DE LONDRINA.	FÁBRICA DE SOFTWARE MANTIDA PELO DEPARTAMENTO DE COMPUTAÇÃO DA UNIVERSIDADE ESTADUAL DE LONDRINA.	Revisão bibliográfica, análise da norma ISO, descritivo	Como resultado deste trabalho, o processo de desenvolvimento de software está pronto para evoluir para o nível E do MPS.Br, oferecendo maior segurança tanto para o projeto quanto para a equipe responsável pela gerência do mesmo.	Como ações futuras para este trabalho, estudos estão sendo realizados para o desenvolvimento de uma aplicação de suporte ao gerenciamento de riscos automatizado. Tal ferramenta irá gerar condições para armazenar as lições 50 aprendidas com os riscos e, conseqüentemente, criar uma memória organizacional de riscos.
PINTO (2011)	APLICAÇÃO DE ARQUITETURA PEDAGÓGICA EM CURSO DE ROBÓTICA EDUCACIONAL COM HARDWARE LIVRE	O objetivo deste trabalho consiste no desenvolvimento e na aplicação de um curso de robótica educacional com elementos de baixo custo e utilização de hardware livre, norteado por uma arquitetura pedagógica interativa.		Revisão bibliográfica em livros e artigos; Desenvolvimento e organização do curso de robótica educacional com hardware livre segundo a arquitetura pedagógica interativa em camadas; Início do estudo de caso com a aplicação do curso para professores da rede pública de ensino.	Colaborou significativamente para a inserção da robótica educacional no cotidiano dos professores participantes.	Desenvolvimento de atividades didáticas remotas com robótica educacional utilizando a plataforma Arduino.
MEDEIROS, ANDRADE, ALMEIDA, ALBUQUERQUE, MEIRA (2004)	Construindo uma Fábrica de Software: da Concepção às Lições Aprendidas	Processo de criação de uma fábrica de software onde foram discutidas, detalhadamente, as etapas de definição e implantação, ressaltando as lições aprendidas durante a execução dos projetos experimentais.		Revisão bibliográfica	A execução dos projetos experimentais foi de grande importância para a apreciação do processo utilizado pela fábrica, permitindo a realização de ajustes necessários, como realocação de participantes a perfis apropriados. O êxito conseguido durante a instanciação do processo adotado, além de servir como fator de motivação para realização de outros projetos.	Realizar projetos de grande porte, a fim de realizar alguns ajustes e refinar ainda mais os processos anteriormente definidos, visando sempre a melhoria contínua.

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
TOMOMITSU, CAMARGO, FILHO (2006)	Dimensões a considerar na análise dos problemas de ensino e aprendizagem de engenharia de software	Focalizar a taxionomia do Processo de Ensino e Aprendizagem de Engenharia de Software, por meio da análise das dimensões da Estrutura Curricular, da Gestão de Currículo, do Ensino e da Aprendizagem, para contribuir com a comunidade científica e identificar linhas de pesquisa na área.	Processo de aprendizagem em Engenharia de Software	Revisão bibliográfica	Ensinar focado no aprendiz é fundamental para desenvolvimento das competências profissionais. O detalhamento de uma epistemologia da prática de Engenharia de software e de estudos que aumentem a significância desse Ensino para os alunos torna-se linha de investigação a ser aprofundada.	
BRITO, SILVA, CABRAL (2013)	Elaboração de uma metodologia de desenvolvimento de software para a fábrica de software de uma instituição de ensino	Formulação de uma metodologia padrão que seja adequada para as necessidades do instituto oferecendo as melhores condições de trabalho para as equipes envolvidas na Fábrica de Software do Câmpus Inhumas.	Adaptar as metodologias conceituadas encontradas na literatura, especialmente voltadas ao mercado de trabalho, para o meio acadêmico. Atender necessidades internas da instituição.	Revisão bibilográfica	Proporcionar aos estudantes a experiência prática de ensinamentos e conhecimentos teóricos vistos em sala de aula, como o caso de técnicas e metodologias de engenharia de software presentes na MDS-FSW assim como ajudar a GTI do Câmpus Inhumas em implementar projetos que facilitam as atividades rotineiras do câmpus.	Revisão do cálculo das horas de estimativa dos casos de uso e as reuniões diárias. Utilização de protótipos para apoiar a coleta de requisitos, diminuindo as incertezas e aumentando a produtividade, utilização de padrões de programação, melhorando a documentação e agilidade da fase de programação e manutenção dos sistemas, estudo e análise do processo unificado aberto denominado Openup e estudo e análise do Guide to the Software Engineering Body of Knowledge (SWEBOK).
SANTOS, SETTE, FILHO, RAMOS, SOUZA, LIMA, BACELAR, CARVALHO, SILVA (2010)	Experiência Acadêmica de uma Fábrica de Software utilizando Scrum no Desenvolvimento de Software	Apresentar um experimento acadêmico, que consiste na vivência de uma fábrica de software na disciplina de Engenharia de Software do curso de pós-graduação em Ciência da Computação, baseado no método Scrum, em um processo de Desenvolvimento Distribuído de Software.	Aplicado à disciplina de Engenharia de Software do curso de pós-graduação em Ciência da Computação	Estudo de Caso; Revisão bibliográfica	A principal contribuição deste artigo está em uma detalhada descrição da vivência de uma fábrica de software visando os aspectos positivos e negativos no desenvolvimento do software utilizando a metodologia Scrum.	Dar continuidade ao projeto FireScrum a fim de realizar alguns ajustes e refiná-los, visando sempre a otimização do projeto, bem com sua melhoria contínua e difusão para empresas ou modelos de negócio.
QUINTELA, ALMEIDA (2006)	Fábrica de Software: análise do impacto na competitividade	Identificar o impacto da implantação de estruturas de Fábrica de Software, por empresas prestadoras de serviço de desenvolvimento de software, quanto ao ganho de competitividade		Foi realizada uma pesquisa de campo junto a empresas prestadoras de serviço de desenvolvimento de software, em estruturas de Fábrica de Software.	Conclui-se a implementação de estruturas de Fábrica de Software pode ser utilizada para transformar a Cadeia de Valor.	

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
CASTOR (2006)	Fábrica de Software: Passado, Presente e Futuro	Contribuir de alguma forma para uma compreensão mais consistente sobre o modelo de fábrica de software e sobre este novo momento de sua expansão pelo mundo.		Revisão bibliográfica	Aponta uma tendência de expansão das Fábricas de Software.	
OSIAS (2008)	FÁBRICA DE SOFTWARE: um estudo de caso na Dataprev, sob a ótica da estrutura organizacional	O objetivo deste estudo de caso foi investigar como o processo evolutivo do desenvolvimento do software, notadamente a mudança de paradigma para a fábrica de software, afetou a estruturação da Diretoria de Relacionamento, Desenvolvimento e Informações – DRD da Dataprev, empresa pública de tecnologia e informações da Previdência Social, refletindo (ou não) um retorno ao modelo taylorista-fordista de organização da produção.	O estudo não consistiu em uma análise histórica, pois tratou apenas da transição ao contexto atual da estrutura organizacional, e nem se propôs a fazer prescrições sobre a mudança e sobre o processo de desenvolvimento de software adotado, já que esta pesquisa teve caráter descritivo e explicativo. Possíveis alterações no clima e cultura organizacional decorrentes de mudanças ocorridas, bem como a reação dos trabalhadores e clientes também não foram tratadas neste estudo.	Para tal foram realizadas entrevistas e aplicado um questionário, além de revisão documental e bibliográfica, sobre os seguintes temas: estrutura organizacional, fábrica de software, taylorismo e fordismo.	A partir dos dados levantados e analisados, sob a perspectiva estudada, entendeu-se que não há um retorno ao modelo taylorista-fordista de organização da produção no processo de adoção do conceito de fábrica de software na Dataprev. E mais: observou-se uma flexibilização na estrutura organizacional de suas unidades de desenvolvimento, as fábricas de softwares da Dataprev.	Como sugestão de futuros trabalhos, foi apontado o aprofundamento do estudo de caso DRD/Dataprev, sob outras e diversas perspectivas; a experiência da Dataprev na contratação e relacionamento com uma fábrica de software externa; e a própria evolução e resultados alcançados por outras empresas no mercado nacional a partir da adoção deste novo conceito no processo de desenvolvimento de software.
MARQUES, RAMOS, SILVA, MACIEL	Fábricas de Software e o processo de desenvolvimento segundo a experiência da FábricaUm	Apresentar a experiência de construção de uma fábrica de software, a FábricaUm, criada como projeto de uma disciplina do mestrado em Ciência da Computação da Universidade Federal de Pernambuco.		Revisão bibliográfica	Apresentou a experiência da FábricaUm, enquanto fábrica de software, na definição de um processo de desenvolvimento e na execução deste processo em seus projetos.	Gerenciamento de desenvolvimento remoto, com produtividade e qualidade aceitável pelo mercado.
GUEDES (2014)	FATORES QUE INFLUENCIAM NA MIGRAÇÃO DO MPS.BR PARA O CMMI NAS EMPRESAS DE SOFTWARE BRASILEIRAS	A pesquisa busca identificar os fatores que influenciam a migração do modelo MPS.BR para o CMMI.	Empresas brasileiras que migraram do MPS.BR para o CMMI.	Os métodos de coleta de dados utilizados para entender o fenômeno da migração foram: o cruzamento da lista de avaliações das empresas do MPS.BR e CMMI, revisão não sistemática da literatura, pesquisa de campo aplicada junto ao grupo de implementadores e avaliadores do MPS.BR e outra realizada nas empresas que realizaram o processo de migração entre os modelos citados.	Concluiu-se que o modelo MPS.BR é capaz de atender e se adequar às necessidades das empresas de software do mercado nacional, mas a maioria das organizações pretende expandir internacionalmente e por isso migram para o CMMI.	É possível complementá-lo com a realização da análise da migração do MPS.BR para outros modelos e normas de melhoria no processo de software adotados por empresas de software no Brasil. Propõe-se ainda, para trabalhos futuros, a realização de uma revisão sistemática criteriosa, com o objetivo de mapear e identificar na literatura os fatores críticos de sucesso envolvidos na implementação, avaliação e continuidade do modelo MPS.BR.

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
DORIGAN (2010)	Gerenciamento de Requisitos: Um Comparativo entre Metodologias Tradicionais e Ágeis sob a ótica dos Modelos de Qualidade.	Apresentar os conceitos e o gerenciamento de requisitos na metodologia tradicional RUP (Rational Unified Process) e na metodologia ágil Scrum, mostrar um comparativo entre Casos de Uso e User Story, suas semelhanças e diferenças no que diz respeito à como os requisitos são levantados, como são mantidos, e qual a participação do cliente nessa parte do desenvolvimento.	Metodologias Ágeis de Desenvolvimento de Software.	Revisão bibliográfica	Apresentação de como é feito o gerenciamento de requisitos no desenvolvimento de software, utilizando metodologias tradicionais, em específico o RUP, e metodologias ágeis, em específico o Scrum, aliado com os modelos de qualidade CMMI e MPS-Br, em seus níveis que tratam os requisitos.	Um banco de dados que armazene os artefatos completos e validados, e que permita a recuperação dos requisitos para uma reutilização seria de grande importância, já que essa área em específico, ainda oferece poucas opções de ferramentas ou algo que auxilie o levantamento de requisitos com o cliente.
FERRARINI (2006)	IDENTIFICAÇÃO E VALORAÇÃO DE COMPETÊNCIAS PARA O DESENVOLVEDOR DE SISTEMAS DE INFORMAÇÃO, NA VISÃO DOS GESTORES DE FÁBRICA DE SOFTWARE DE SALVADOR	O objetivo deste estudo foi identificar e valorar as competências desejadas por gestores no perfil dos desenvolvedores de sistemas, em especial os que se encontram em início de carreira, em fábricas de software de Salvador, estado da Bahia.		Estudo de caso: 1. levantar, definir ou caracterizar competências desejáveis à funções específicas; 2. diagnóstico das necessidades de novas competências tendo em vista a perspectiva de novas situações de trabalho como a introdução de sistemas de gestão; ou 3. pesquisas realizadas para identificar que tipo de competências gerenciais são necessárias a um certo tipo de segmento produtivo. Para a coleta de dados, o instrumento escolhido foi a entrevista.	Como conclusão, o trabalho apresenta as competências identificadas por meio da percepção dos gestores, onde foi verificado que, na fábrica de software, as competências técnicas são as mais valorizadas. Foi possível verificar que realmente, o modelo de fábrica aplica a especialização do trabalho. Isto significa delimitar as tarefas, atividades e funções que cada um deve e pode realizar. O processo conduz tudo, e as pessoas são guiadas por ele.	Definição das competências organizacionais para o modelo de fábrica de software. A partir daí, seria possível detalhar grupos de competências individuais necessários, podendo servir como ferramenta de apoio ao gestor da fábrica de software na busca de um perfil de profissional que se encaixe nas necessidades reais da equipe.

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
VERGARA (2014)	Implantação de Softwares como Serviço em uma Nuvem Privada para a Fábrica de Software da FGA	O objetivo deste trabalho é trazer ao leitor primeiramente o estado da arte sobre computação em nuvem, posteriormente, uma breve indicação de ferramentas para computação em nuvem, passando por todos os níveis de arquitetura, e posteriormente a proposição de um modelo de implantação de computação em nuvem para a fábrica de software da faculdade do Gama e por último, a implantação de parte deste modelo.		A metodologia de pesquisa proposta foi dividida em três grandes fases. Primeiramente uma fase para levantamento bibliográfico e estudo sobre computação em nuvem, o que é, quais seus tipos, como é sua arquitetura e etc. Posteriormente passou-se para uma fase de aplicar esse conhecimento adquirido para um estudo de caso real, no caso a fábrica de software do gama, nesta fase foi feita uma avaliação de que tipo de serviços e qual os respectivos softwares que beneficiariam a fábrica nas suas atividades, pensou-se em soluções entre as três principais áreas da computação em nuvem, e por dentre todas as ferramentas analisadas foi escolhida a solução que trouxesse o melhor custo benefício para a fábrica, tendo como parâmetros facilidade de implantação X relevância X tempo para implantação.	Ficou evidente neste trabalho que a implantação de serviços em uma nuvem privada não é um trabalho simples e que precisa de muita dedicação e paciência, principalmente se você deseja que ela esteja cada vez mais segura. Neste trabalho é mostrado algumas coisas que podem ser feitas para que a segurança seja aumentada, como o acesso pela terceira parte (VPN) mascarando os servidores da rede, e a proteção por meio de firewalls.	O primeiro é terminar a implantação do SaaS proposto no modelo de implantação para a fábrica. Para terminar esta implantação falta a criação de um servidor de SMTP para que o Expresso tenha seu funcionamento de troca de emails funcional, e também da instalação de um servidor de BigBlueButton para a instalação do módulo de vídeo conferência do Expresso. Desta maneira a implantação do SaaS pode ser considerada completa. Um segundo ponto para trabalho futuro é a implantação do IaaS e do PaaS, previstos no modelo de de implantação para a fábrica. Estes serviços não foram implantados por não serem tão importantes frente aos serviços de SaaS e por haver pouco tempo para a implantação.
PESSOA (2009)	PROCESSOS E PROJETOS EM UMA FÁBRICA DE SOFTWARE ELAB-TI	Aplicar modelos de processo e metodologias de desenvolvimento no ambiente de fábrica e avaliar seus resultados; Descrever um processo padrão de desenvolvimento de software para fábricas de software; Identificar e estabelecer diretrizes para os processos críticos de uma fábrica; Estabelecimento de um processo de medição de processo para fábrica de software; Desenvolver métodos e técnicas próprios para o ambiente de fábrica, fortemente calcados em automação e reuso.		A análise estruturada que procura identificar os dados existentes nas operações do mundo real, estruturá-los e representar seu fluxo e operações.	A principal contribuição oferecida pela pesquisa é o estabelecimento de diretrizes estratégicas para pequena empresa sempre preocupada com competitividade, à luz de um referencial teórico clássico presente somente nas grandes empresas. Outra contribuição importante foi o processo de condução da pesquisa-ação.	

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
BORGES, CARVALHO, MORAES (2012)	Programa de Extensão “Fábrica de Software Acadêmica”: contribuindo para a formação profissional na área da Informática	Capacitar os alunos para a compreensão e resolução de problemas relacionados à produção de software, seguindo processos de desenvolvimento de sistemas; Promover a auto regulação das aprendizagens discentes; Pesquisar e aplicar novas tecnologias e metodologias de desenvolvimento de software; Divulgar o potencial dos alunos participantes junto ao mercado de trabalho, com vistas a obtenção de colocações dentro das empresas de desenvolvimento de software; Dar suporte a outros projetos do instituto, fornecendo soluções de software personalizadas; Promover ações de extensão (palestras, cursos de formação complementar, eventos comunitários, entre outros).	Ambiente acadêmico	Revisão bibliográfica	Os benefícios desta ação de extensão, tanto para alunos e professores, quanto para a própria instituição, podem ser observados na medida que: o aluno tem a oportunidade de praticar os conteúdos vistos em sala de aula, aprimorando e expandindo seus conhecimentos; o aluno pode realizar o seu estágio obrigatório ou trabalho de conclusão de curso dentro da Fábrica; o aluno permanece mais tempo dentro da instituição, com aula em um dos turnos e no outro colaborando com o projeto; a instituição reduz as chances de evasão nos cursos e os professores pesquisadores contam com o apoio da equipe da Fábrica de Software na execução das demandas de seus projetos.	

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
CRUZ, QUISPE, SUCUPIRA, LEONARDO, MATHEUS, MONSORES, YAGUI, CHAN, LIMA (2013)	Relato de um experimento piloto de uma Fábrica de Software baseada em métodos ágeis	Este trabalho tem como objetivo apresentar o relato de um experimento piloto de uma FSMA (Fábrica de Software baseada em Metodologias Ágeis) centrada no processo ensino-aprendizagem (PEA) e que adotou métodos ágeis como metodologia de desenvolvimento de produtos de software.		Os experimentos foram realizados de modo colaborativo, criando um espaço de referência, onde as funcionalidades dos dois sistemas Web foram estruturadas em função dos saberes previamente adquiridos ao longo do curso de graduação pelos petianos e das necessidades das comunidades de alunos do curso de graduação. Os experimentos desenvolvidos pelos alunos foram compostos pelas seguintes etapas da Engenharia de Software: análise e levantamento de requisitos dos (dois) sistemas; definição da arquitetura dos sistemas; implementação; testes de funcionalidades e implantação na infra-estrutura de servidores Web administrativos já disponíveis na Universidade.	<p>A FSMA se estabeleceu sob princípios de desenvolvimento centrados em PEA, buscou-se: Priorizar o indivíduo (aluno) e suas interações ao invés do foco no desenvolvimento de processos formais e uso intensivo de ferramentas de modelagem; Desenvolver software executável ao invés de documentação; Produzir respostas rápidas a mudanças nos requisitos ao invés de seguir planos rígido.</p> <p>A associação dos MA, programação XP e do PEA na FSMA adotada pelo grupo PET-SI neste experimento foi bastante proveitosa. Apresentou as seguintes vantagens: Diluiu o atual paradigma da prática educativa comum a muitos cursos de Sistemas de Informação que apregoam a massificação da formação de alunos-desenvolvedores com pouca ou nenhuma capacidade de resolução de problemas práticos e reduzido poder crítico-reflexivo; Apresentou aos petianos, o princípio da indissociabilidade entre ensino, pesquisa e extensão, apresentando de forma prática a articulação entre essas três atividades; Desenvolveu uma atividade acadêmica, em padrões de qualidade e excelência, voltada para os alunos do curso de Sistemas de Informação, contribuindo para a elevação da qualidade da formação dos petianos estimulando seu espírito crítico e reduzindo as distâncias entre a teoria da computação e a sua prática.</p>	Refinamentos na FSMA e nos dois produtos. Além disso, novos experimentos serão realizados.
BOENTE, OLIVEIRA, ALVES (2005)	RUP como Metodologia de Desenvolvimento de Software para Obtenção da Qualidade de Software	Discussão conceitual sobre qualidade, inicialmente. Discutimos os principais e relevantes aspectos do método RUP, defendendo-o, em conjunto com a Unified Modeling Language (UML), para uso na construção de softwares.		Revisão bibliográfica	Recomendamos a implementação de projeto de RUP como metodologia de desenvolvimento de software para obtenção da qualidade do produto de software gerado.	

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
SILVA (2011)	Sistema de Gerenciamento de Tarefas para usuários de Scrum	O trabalho apresenta a idealização, projeto e implementação do iPlan, um sistema voltado para usuários da metodologia ágil Scrum, cujo principal objetivo é auxiliar no gerenciamento de tarefas necessárias ao completo desenvolvimento de um ou mais projetos.	Utilização do Scrum.	Revisão Bibliográfica	A idealização, projeto e implementação do iPlan, um sistema cujo objetivo é facilitar a vida de usuários da metodologia ágil Scrum, principalmente Product Owners e ScrumMasters no que diz respeito ao gerenciamento de tarefas necessárias para a elaboração de um projeto.	Desenvolvimento de novas funcionalidades afim de tornar o iPlan adaptável a diferentes equipes sem que seja necessária manipulação de seu código fonte.
FILHO (2011)	Um estudo analítico entre as abordagens de Engenharia de Requisitos nas Metodologias Ágeis XP, SCRUM e Crystal	Estudo analítico entre as metodologias ágeis Crystal, Scrum e XP visando possíveis formas de obter maior eficiência no desenvolvimento de software. Por meio de uma divisão de parâmetros qualitativos focados na engenharia de requisitos de cada processo analisado adequa as abordagens a serem incorporadas no desenvolvimento de organizações.	Metodologias Ágeis de Desenvolvimento de Software.	Revisão Bibliográfica	Nas três metodologias apresentadas e estudadas percebe-se que a comunicação entre cliente e equipe de desenvolvimento é um elemento essencial para a progressão do projeto. Não existe uma solução ideal para qualquer situação como a engenharia de software sempre nos mostrou, mas a adoção de modelos vai depender muito dos fatores humanos e de projeto como estudados.	
SATO (2007)	Uso eficaz de métricas em métodos ágeis de desenvolvimento de software	Investigar o uso de métricas no acompanhamento de projetos utilizando Métodos Ágeis de Desenvolvimento de Software		As métricas usadas neste estudo de caso são métricas de diagnóstico: Total de Linhas de Código, Total de Linhas de Código de Teste, Número de Linhas Alteradas, Número de Commits, Estimativas Originais, Estimativas Finais, Número de Histórias Entregues, Número de Pontos Entregues, Tempo, Tamanho da Equipe, Esforço, Complexidade Ciclomática de McCabe, Métodos Ponderados por Classe, Falta de Coesão de Métodos, Profundidade da Árvore de Herança, Número de Filhos, Acoplamento Aferente, Acoplamento Eferente. As métricas quantitativas e objetivas foram coletadas de forma automatizada das seguintes fontes: Plug-in do Eclipse, Repositório de Código, XPlanner.	Estudou o papel das métricas no acompanhamento de projetos utilizando Métodos Ágeis de desenvolvimento de software.	Conduzir mais estudos para validação de outras métricas de acompanhamento e, principalmente, métricas organizacionais. Estudar melhor esse processo de definição, coleta e adaptação das métricas.

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
AMÂNCIO, COSTA, CAMARGO, PENTEADO (2009)	Gerência de Recursos Humanos para uma Fábrica de Software de Pequeno Porte	Propõe um processo de gerência de recursos humanos para uma fábrica de software de pequeno porte de uma instituição pública de ensino superior.	Alta rotatividade de pessoal.	Revisão Bibliográfica		Construção de templates para os artefatos de software proposto
FABRI, BEGOSSO, PESSOA, SPÍNOLA (2006)	Desenvolvimento do Conceito sobre Fábrica de Software em Instituições de Ensino que possuem Cursos de Computação	Embutir o conceito sobre fábrica de software aos alunos que cursam a disciplina de Engenharia de Software; Desenvolver um modelo de ensino de engenharia e software que prime por questões fabris como qualidade e produtividade. É importante salientar que esse modelo pode ser instanciado por qualquer instituição de ensino; Desenvolver um laboratório de desenvolvimento de software, que utilize um processo fabril em sua arquitetura. Nesse laboratório os alunos poderão visualizar na prática os conceitos sobre fábrica de software abordados em sala de aula.		Primeira Fase - Pesquisa-ação Segunda Fase - Experimental	Apresentou os resultados obtidos pelo projeto “desenvolvimento do conceito sobre fábrica de software em instituições de ensino que possuem cursos de computação”. Uma das justificativas para discussão sobre Fábrica de Software e Engenharia de Software no processo de ensino e aprendizagem está relacionada a questões de qualidade e produtividade.	Configurar o laboratório de engenharia de software.
OLIVEIRA, NETO (2003)	Fábrica de Software: Promovendo a Criação de Empresas Competitivas em Tecnologia da Informação	Descreve as experiências desenvolvidas no projeto fábrica de software em implantação na Faculdade de Tecnologia de Jundiaí-SP do Centro Paula Souza.		Estudo de Caso	Conclui-se que esta é uma maneira altamente inovadora e que apresenta resultados efetivos na formação de futuros profissionais desenvolvedores de sistemas informatizados, além de ser a melhor forma na busca de uma real democratização na utilização da Tecnologia da Informação.	

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
CAMPELLO, SILVA (2011)	IDENTIFICAR ETAPAS CONVERGENTES ENTRE O MÉTODO CLEANROOM E A METODOLOGIA ÁGIL SCRUM	Apresentar pontos de convergência entre a metodologia Scrum e o método Cleanroom no processo de desenvolvimento de softwares e gerenciamento de projetos em ambientes instáveis.		Estudo de Caso e Revisão Bibliográfica	Identificou-se que a expectativa dos profissionais de desenvolvimento quanto a qualidade do software aumenta com a utilização das técnicas propostas pelo Cleanroom e pelo Scrum. Outro fator importante está sobre o gerenciamento de equipes em ambientes mutáveis, os dados mostram que reuniões diárias de alinhamento, as stand-up meetings do Scrum, e o desenvolvimento estruturado, utilizado no Cleanroom, permitem alterações mais suaves entre recursos na equipe estando estes no mesmo nível de competência ou em diferentes níveis de competência.	
VIVACQUA (2009)	FÁBRICAS DE SOFTWARE E A ACADEMIA: ANÁLISE DA FORMAÇÃO ACADÊMICA EM INFORMÁTICA NO MUNICÍPIO DO RIO DE JANEIRO	Analisar se as principais instituições de ensino nas áreas de Ciências da Computação, Informática e Engenharia de Computação formam profissionais adequadamente capacitados, vis-à-vis as necessidades inerentes às organizações do tipo “Fábrica de Software”.	O estudo delimitou-se a instituições de ensino superior nas áreas de Ciência da Computação, Engenharia de Sistemas e Informática oferecidas no município do Rio de Janeiro.	Revisão Bibliográfica e Entrevista	Conclui-se, portanto, que as grades curriculares são adequadas considerando-se um papel formativo geral dos cursos superiores. A complementação das 109 competências por meio do treinamento em ferramentas de uso profissional deve ser obtida por meio do estágio, de cursos extracurriculares, cadeiras optativas ou eletivas, cursos de especialização ou do aprendizado no trabalho.	A primeira sugestão é o estudo do desempenho do aluno recém-graduado nas funções de sua atividade profissional, ou seja, uma avaliação a posteriori sobre sua graduação. Sugere-se, também, o aprofundamento do estudo em duas dimensões: a primeira é pesquisar um maior número de instituições, em todo o país; a segunda é a realização de estudos de caso das instituições que possam produzir um retrato mais fiel da visão da academia em relação às Fábricas de Software, nos quais todos os professores que participam da elaboração das grades curriculares sejam ouvidos e que uma amostra mais significativa dos alunos seja estudada.
BRITO, FERREIRA, SILVA, BURÉGIO, LEITE (2004)	Uma experiência na implantação de processo em uma fábrica de software livre	Apresentar um experimento acadêmico baseado no conceito de fábrica de software [Aaen, 1997] distribuída utilizando um processo de desenvolvimento adaptado para o modelo OSS.	Foram percebidas algumas dificuldades na utilização do processo, principalmente porque o mesmo estava muito sobrecarregado de atividades e a fábrica não possuía pessoal disponível para sua execução. Devido a isso, alguns processos não foram completamente executados.	Pesquisa-ação	Apresentou a experiência de uma equipe formada por estudantes e Profissionais de TI na concepção de uma Fábrica de Software Livre, onde houve a definição de um processo de software e desenvolvimento de projetos para um cliente real.	

Autor	Título	Objetivo	Limitação	Método	Resultados	Trabalhos futuros
FILHO (2008)	FÁBRICA DE SOFTWARE: UM ESTUDO DE CASO, SOB A ÓTICA DA FLEXIBILIZAÇÃO ORGANIZACIONAL E DAS RELAÇÕES DE TRABALHO	Levantar a situação atual das fábricas de software no Brasil com relação aos conceitos de flexibilização organizacional e de flexibilização das relações de trabalho, por meio do estudo da bibliografia disponível, e de um estudo de caso em uma fábrica de software escolhida.	Este estudo não consistiu em uma revisão histórica, pois tratou apenas da situação no contexto atual da estrutura organizacional. Também não se propôs a fazer prescrições sobre mudanças das relações de trabalho e nem sobre o processo de desenvolvimento de software adotado, já que esta pesquisa tem caráter descritivo e explicativo.	Revisão Bibliográfica e Estudo de Caso	Contribuir para o entendimento de como são as relações de trabalho em empresas de fábrica de software de grande porte.	As sugestões a seguir dividem-se em dois grupos: 1. Pesquisas que aperfeiçoem as conclusões obtidas neste trabalho por meio de análises mais detalhadas; 2. Desenvolvimento de temas surgidos no desenvolver deste trabalho.
LEITE, LUCRÉDIO (2014)	Desenvolvimento de Software utilizando o Framework Scrum: um Estudo de Caso	Apresentou um estudo sobre a aplicação do Scrum no desenvolvimento de um software para informatizar os processos de gestão de estoque do Restaurante Universitário da Universidade Federal de São Carlos, realizados a princípio de forma manual.		Estudo de Caso	Concluiu-se então que a principal contribuição deste trabalho refere-se à identificação de empecilhos e possíveis soluções ao implantar o Scrum, tendo como colaboradores elementos inexperientes no que concerne a abordagem.	Projetos futuros podem estender a utilização do framework para outras áreas da universidade, investigando quais os contratemplos encontrados e compará-los aos apresentados neste artigo.
SOARES, MARIZ, CAVALCANTI, RODRIGUES, NETO, BASTOS, ALMEIDA, PEREIRA, ARAÚJO, CORREIA, ALBUQUERQUE (2007)	Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software	Tem por objetivo relatar as experiências obtidas na adaptação de um processo de desenvolvimento distribuído com base no SCRUM realizado por uma fábrica de software open source.		Revisão Bibliográfica e Estudo de Caso	Apresentou um estudo de caso sobre a aplicação de métodos ágeis, particularmente baseado na abordagem proposta pelo Scrum, em um processo para desenvolvimento de software open source com fortes características de desenvolvimento distribuído.	

**ANEXO A – EMENTA DA DISCIPLINA “METODOLOGIA DE
DESENVOLVIMENTO DE SISTEMAS” DO CURSO DE SISTEMAS DE
INFORMAÇÃO**

ASSOCIAÇÃO EDUCACIONAL DOM BOSCO
FACULDADE DE CIÊNCIAS ECONÔMICAS, ADMINISTRATIVAS
E DA COMPUTAÇÃO DOM BOSCO
Seção Técnica de Ensino

2016

4º Ano

PROGRAMA DE DISCIPLINA

CURSO: SISTEMAS DE INFORMAÇÃO

DISCIPLINA: METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

NÚMERO DE AULAS TEÓRICAS: 60

NÚMERO DE AULAS PRÁTICAS: 20

CARGA HORÁRIA TOTAL : 80

EMENTA

Principais metodologias de desenvolvimento de Sistemas existentes, entre elas:

- CMM – Capability Maturity Model
- CMMI – Capability Maturity Model Integration
- MPS.BR
- RUP – Rational Unified Process
- XP – Extreming Programming
- SCRUM

OBJETIVOS GERAIS:

Demonstrar, para os alunos, a influência que processos de engenharia de sistemas têm em projetos de criação de Sistemas de Informação;

- 1) Analisar o estado da arte da Engenharia de Software.
- 2) Descrever as principais metodologias atuais usadas pela Engenharia de Software.
- 3) Definir os principais recursos para o desenvolvimento organizado de sistemas de software.
- 4) Identificar, descrever e comparar os modelos de processo de desenvolvimento de software.

ASSOCIAÇÃO EDUCACIONAL DOM BOSCO		
ESTRUTURAÇÃO DO CONTEÚDO DA DISCIPLINA: METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS		
Nº DA UNID.	ASSUNTO	Nº DE HORAS-AULA
1	INTRODUÇÃO A METODOLOGIA DE DESENVOLVIMENTO 1.1 Introdução. 1.3 Diferenças entre um setor organizado e um setor desorganizado.	06
2	CMM – CAPABILITY MATURITY MODEL 2.01 – INTRODUÇÃO AO CMM. 2.02 – NIVEL 1. 2.03 – NIVEL 2. 2.04 – NIVEL 3. 2.05 – NIVEL 4. 2.06 – NIVEL 5.	14
3	CMMI – CAPABILITY MATURITY MODEL INTEGRATION 3.01 – DIFERENÇAS COM O CMM. 3.02 – LABORATÓRIO DE CMM.	06
4	MPS.BR – MELHORIA DE PROCESSO DO SOFTWARE BRASILEIRO 4.01 – DIFERENÇAS COM O CMMI. 4.02 – LABORATÓRIO DE MPS.BR.	06
5	RUP – RATIONAL UNIFIED PROCESS 5.01 – INTRODUÇÃO AO RUP. 5.02 – FASES DO RUP. 5.03 – APLICAÇÃO DO RUP NO DESENVOLVIMENTO DE SISTEMAS. 5.04 – LABORATÓRIO DE RUP	14
6	XP – EXTREMING PROGRAMMING 6.01 – INTRODUÇÃO A METODOLOGIA ÁGIL 6.02 – PILARES DO XP 6.03 – UTILIZANDO XP NO DESENVOLVIMENTO DE SISTEMAS 6.04 – LABORATÓRIO DE XP	14
7	SCRUM 7.01 – PILARES DO SCRUM 7.02 – UTILIZANDO SCRUM NO DESENVOLVIMENTO DE SISTEMAS 7.03 – LABORATÓRIO DE SCRUM	20

ASSOCIAÇÃO EDUCACIONAL DOM BOSCO

**MÉTODOS E TÉCNICAS DE ENSINO EMPREGADOS NA DISCIPLINA DE
METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS**

Metodologia:

AULAS TEÓRICAS

Aulas expositivas sobre os princípios teóricos com auxílio de slides, transparências e modelos e práticas, de desenvolvimento em paralelo, junto com os alunos, com auxílio de um projetor.

AULAS PRÁTICAS

Aulas no laboratório de informática.

Sistema de Avaliação:

O sistema de avaliação da disciplina será constituído de seis provas anuais, sendo quatro provas bimestrais, uma prova de exame final e uma prova de segunda época.

As avaliações escritas serão compostas de questões dissertativas e objetivas – quando se tratar de conteúdos qualitativos; e dissertativas, objetivas e com exercícios – quando se tratar de conteúdos quantitativos.

BIBLIOGRAFIA BÁSICA:

- BEZERRA, Eduardo. **Princípios de análise e projetos de sistemas com UML**. 2.ed. Rio de Janeiro: Elsevier, 2007.
- PRESSMAN, R. S. **Engenharia de software**. 7.ed. São Paulo: Makron Books do Brasil, 2011.
- SCHACH, Stephen R. **Engenharia de software: os paradigmas clássico & orientado a objetos**. 7.ed. São Paulo: McGraw Hill, 2009.
- KOSCIANSKI, André. **Qualidade de Software** – 2.ed. São Paulo: Novatec, 2007

BIBLIOGRAFIA COMPLEMENTAR:

- MANZANO, José Augusto Navarro Garcia. **Estudo dirigido de Microsoft Visual C# 2012 express**. São Paulo: Érica, 2010.
- YOURDON, Edward. **Análise estruturada moderna**: tradução da terceira edição americana. Rio de Janeiro: Campus, 1990.
- COAD, Peter; YOURDON, Edward. **Análise baseada em objetos**. Rio de Janeiro: Campus, 1991.
- REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. 2.ed. São Paulo: Brasport, 2002.
- RESENDE, Antônio Maria Pereira de; SILVA, Claudiney Calixto da. **Programação orientada a aspectos em java**: desenvolvimento de software orientado a aspectos. Rio de Janeiro: Brasport, 2005.