

Instituto de Pesquisas Tecnológicas do Estado de São Paulo

Herez Moise Kattan

**Técnica de desenvolvimento de sistemas baseada na fusão entre
engenharia simultânea e programação em pares.**

**São Paulo
27/05/2014**

Herez Moise Kattan

Técnica de desenvolvimento de sistemas baseada na fusão entre engenharia simultânea e programação em pares.

Exame de Qualificação apresentado ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Computação.

Área de Concentração: Engenharia de *Software*.

Orientador: Prof. Dr. José Eduardo Z. Deboni

São Paulo
27/05/2014

RESUMO

A programação em pares é a principal prática do método XP (*Extreme Programming*). Ela melhora a qualidade e reduz o número de defeitos do sistema. Aumenta a retenção, a motivação e a confiança entre os membros da equipe. Porém seu ônus é a perda de produtividade e este problema é o principal objeto de estudo desta dissertação. A melhoria da qualidade da programação em pares está diretamente relacionada ao nível de experiência dos programadores e complexidade da tarefa. Quão mais experiente é o programador e menos complexa for a tarefa, menor será a melhora da qualidade e redução do número de defeitos. Este trabalho usa a engenharia simultânea, visando tornar a programação em pares adequada para programadores experientes e tarefas complexas. No começo desta dissertação, é elaborada uma técnica de desenvolvimento de sistemas, com intento de aprimorar e unir, a melhoria da qualidade e redução do número de defeitos da programação em pares ao aumento da produtividade da engenharia simultânea. Depois, são definidas as medidas e métricas, para avaliação da produtividade e qualidade da nova técnica usando como referência a programação em pares. Depois, é realizada experimentação prática em ambiente controlado e estudos de caso em ambiente corporativo. No final, é avaliada a eficácia da nova técnica.

Palavras Chaves: Programação em Pares; Engenharia Simultânea ou Concorrente; Extreme Programming; Métodos Ágeis; Métodos de Desenvolvimento de Sistemas.

1. INTRODUÇÃO

1.1 Motivação

A programação em pares é a principal prática do método XP (*Extreme Programming*). Ela melhora a qualidade e reduz o número de defeitos do sistema. Aumenta a retenção, a motivação e a confiança entre os membros da equipe. Porém seu ônus é a perda de produtividade, este problema é o principal objeto de estudo desta dissertação e não há publicação usando a engenharia simultânea para resolvê-lo. A melhoria da qualidade está diretamente relacionada ao nível de experiência dos programadores e complexidade da tarefa. Quão mais experiente é o programador e menos complexa for a tarefa, menor será a melhoria da qualidade e redução do número de defeitos, também não há publicação usando a engenharia simultânea visando tornar a programação em pares adequada para programadores experientes e tarefas complexas.

A pesquisa conduzida por Smith et al. (2002) na universidade americana West Virginia e NASA, reconheceu o papel fundamental de uma das práticas: a programação em pares. Ela relaciona-se diretamente a quase todas as outras práticas do método XP. A centralidade da programação em pares para o método XP é evidenciada devido ao fato de que caso removida afetaria diretamente todas as práticas relacionadas. Constatou-se o problema de produtividade da programação em pares e foi concluído que a NASA não deveria usar método ágil baseado em programação em pares em 64% dos seus projetos. Para os demais casos, o método usado na pesquisa apenas endossou vantagem sobre abordagens convencionais em um relativamente pequeno e específico conjunto de casos, descritos a seguir:

- O projeto é relativamente pequeno;
- Existe uma abundância de desenvolvedores;
- É necessário um período de desenvolvimento curto.

Segundo Dyba et al. (2007) não é sempre que ocorre a melhoria da qualidade ao usar a programação em pares, conforme é descrito no Quadro 1. Quão mais

experiente é o programador e menos complexa for a tarefa, menor será a melhoria da qualidade.

Quadro 1 – Diretriz para quando usar a programação em pares

Experiência do programador	Complexidade da tarefa	É vantajoso usar a programação em pares?
Júnior	Simplex	Sim, se o aumento da qualidade é o principal objetivo.
Júnior	Complexa	Sim, se o aumento da qualidade é o principal objetivo.
Pleno	Simplex	Não.
Pleno	Complexa	Sim, se o aumento da qualidade é o principal objetivo.
Sênior	Simplex	Não.
Sênior	Complexa	Não, a menos que a tarefa é complexa demais para ser resolvida por apenas um programador sênior.

Fonte: Dyba et. al., 2007, p. 15

Legenda: Júnior = programador iniciante; Pleno = programador de experiência intermediária; Sênior = programador experiente.

Cockburn et al. (2001) e Williams et al. (2000) usaram programadores iniciantes e constataram que a produtividade ao usar a programação em pares parece ser comparável à de duas pessoas trabalhando de forma independente. As razões sugeridas são que programando em pares discutem o sistema antes de desenvolvê-lo, então provavelmente ocorrerá menos erros e retrabalho. Além disso, o número de erros evitados pela inspeção informal realizada pela programação em pares é tal que, menos tempo é gasto na reparação de *bugs* descobertos durante o processo de testes.

Segundo Parrish et al. (2004), há aumento de produtividade quando o desenvolvimento é realizado de maneira concorrente por meio da programação

individual, quando os programadores trabalham em computadores e partes diferentes do sistema. Porém a qualidade demonstrou-se melhor quando usam a programação em pares. A produtividade foi mensurada pelo número de pontos de função desenvolvido por hora por programador. As equipes foram classificadas em níveis de concorrência(simultaneidade) alto e baixo. A qualidade foi mensurada pelo desvio do padrão de codificação. As equipes estavam no mesmo local físico.

Arisholm et al. (2007) e Parrish et al. (2004) realizaram seus trabalhos com programadores experientes. Eles descobriram que houve uma significativa perda de produtividade em comparação a dois programadores trabalhando sozinhos, havia alguns benefícios de qualidade, mas estes não compensavam totalmente a perda da produtividade da programação em pares. No entanto, segundo eles, a partilha de conhecimento que acontece durante a programação em pares é muito importante, pois reduz os riscos globais para um projeto no caso de algum membro da equipe sair e isso pode justificar o seu uso.

O estudo liderado por Dyba et al. (2007) revisou quinze estudos sobre programação em pares e todos os estudos usaram tarefas de programação como base para a comparação. O número de indivíduos nos estudos oscilou entre 12 e 295. O estudo concentrou-se nos aspectos:

- Duração, tempo do calendário de desenvolvimento.
- Esforço, mensurado em horas de trabalho.
- Qualidade, quanto melhor ficou o produto final.

Ao analisar-se o resultado, constata-se a melhora da qualidade, redução do número de defeitos, porém observa-se também ao usar a programação em pares o aumento do esforço no projeto, se comparada à programação individual.

Wray (2010) apresentou quatro mecanismos para melhorar o desempenho da programação em pares:

1. **Comunicação.** Parte da eficácia da programação em pares é presumivelmente devido à interação entre os programadores. Quando os

programadores falam sobre o problema no momento da dúvida, encontram mais rapidamente a solução, consequentemente há o aumento da produtividade;

2. **Rotação do par.** A fadiga diminui a produtividade e trocar a pessoa do par que digita no teclado durante o expediente, pode aumentar a produtividade. É importante levar em consideração que programadores iniciantes aumentam sua produtividade quando seu par é mais experiente.
3. **Lutar contra práticas ruins de programação.** A pressão do par contribui para melhorar a qualidade da programação, evitando retrabalhos e aumentando a produtividade;
4. **Compartilhar e reconhecer competências.** A descoberta de quem são os especialistas em determinados assuntos e compartilhar o conhecimento, pode aumentar a produtividade.

Wray (2010) sugeriu refazer a pesquisa do Dyba (2007) considerando seus quatro mecanismos e pensar em novos critérios e método para mensurar eficientemente o desempenho, levando em consideração estes mecanismos.

A base do conceito da **Engenharia Simultânea** é melhorar a qualidade e aumentar a produtividade com redução de custos. Ainda não há na literatura o uso deste conceito para aumentar a produtividade e aprimorar a qualidade da programação em pares.

O conceito de **Engenharia Simultânea** é baseado na ideia do processamento paralelo/simultâneo dos processos da empresa, que reduz o tempo de lançamento de um novo produto bem como a melhoria da qualidade (PITHON, 2004, p. 3).

Em 1982 foi iniciado um estudo, conduzido pelo DARPA (*Defense Advanced Research Project Agency* – Agência de Projetos de Investigação Avançada de Defesa), buscando uma forma de aumentar o grau de paralelismo das atividades de desenvolvimento de produtos. O resultado deste trabalho, publicado em 1988, definiu o termo Engenharia Simultânea da seguinte forma:

Engenharia Simultânea é uma abordagem sistemática para o desenvolvimento

integrado e paralelo do projeto de um produto e os processos relacionados, incluindo manufatura e pós-venda. Esta abordagem procura fazer com que as pessoas envolvidas no desenvolvimento considerem, desde o início, todos os custos, prazos e requisitos dos clientes. (Winner et al., 1988 apud (Prasad, 1996))

Nas entrelinhas desta definição, o envolvimento de representantes das várias áreas que agregam conhecimento e experiência ao produto podem e devem ser chamadas a participar. Vem daí a necessidade da formação de equipes interdisciplinares coordenadas e voltadas a um objetivo final: satisfazer as necessidades dos clientes, o que sem dúvida alguma trará retorno financeiro às corporações. Marketing, vendas, assistência técnica, testes, fabricação, engenharia, expedição e demais áreas do conhecimento devem ser envolvidas, se não em dedicação completa, pelo menos parcialmente.

A partir da definição de Winner, o conceito de Engenharia Simultânea tornou-se muito mais abrangente, podendo incluir a cooperação e o consenso entre os envolvidos no desenvolvimento, com o emprego de recursos computacionais. A partir desta definição inicial, muitas outras surgiram:

Engenharia simultânea é uma abordagem sistemática para o desenvolvimento integrado de produtos que enfatiza o atendimento das expectativas dos clientes. Inclui os valores de trabalho em equipe, tais como cooperação, confiança e compartilhamento, de forma que as decisões sejam tomadas, no início do processo, em grandes intervalos de trabalho paralelo incluindo todas as perspectivas do ciclo de vida, sincronizadas com pequenas modificações para produzir consenso (Ashley, 1992 apud Prasad, 1996)

Engenharia simultânea é um ambiente de desenvolvimento, no qual a tecnologia de projeto auxiliado por computador é utilizada para melhorar a qualidade do produto, não somente durante o desenvolvimento, mas em todo o ciclo de vida (Ellis, 1992 apud Prasad, 1996)

Engenharia simultânea é uma metodologia de desenvolvimento de produtos, na qual várias habilidades (*X-abilities*) são consideradas parte do processo de desenvolvimento de produtos (manufatura, serviços, qualidade entre outros). Esses requisitos não servem somente para se atingir as funcionalidades básicas do produto, mas para definir um produto que atenda todas as necessidades dos clientes (Hartley, 1992 apud Prasad, 1996)

A Engenharia simultânea é uma idéia aparentemente simples baseada fundamentalmente nas diferentes formas de como os produtos são concebidos, projetados e produzidos. A ideia é a cooperação mútua entre as pessoas, de modo que a de se obter um melhor desempenho a fim de alcançar com sucesso a meta comum (Causing, 1989)

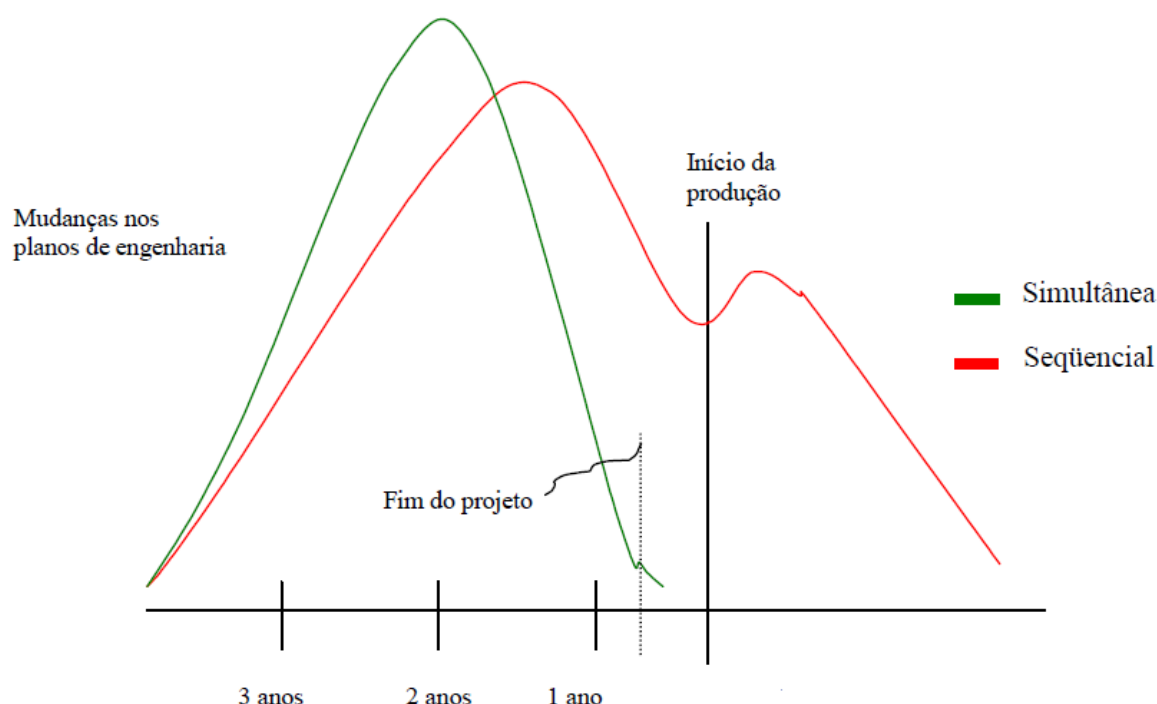
Todas estas definições abordam de alguma maneira várias palavras consideradas chaves para o sucesso da implementação da Engenharia Simultânea, tais como: trabalho em equipe, cooperação, qualidade do produto e ciclo de vida. Estas definições também partilham a hipótese de que a Engenharia Simultânea é o meio para aprimorar a qualidade do projeto do produto com a redução dos custos.

Segundo Bennett et al. (1995) o principal objetivo da Engenharia Simultânea é a diminuição do tempo desde o pedido até a entrega, para um novo produto, com custo mais baixo e maior qualidade. Isto é alcançado através do desenvolvimento paralelo, ao invés de sequencial, das diferentes etapas que compõem o Projeto do Produto, com o emprego de times ou equipes multidisciplinares. Assim como a filosofia do *Just-in-Time*, a Engenharia Simultânea também é uma filosofia e não uma tecnologia. Engenharia Simultânea usa tecnologia para atingir seus objetivos.

Segundo Hartley (1992) na Engenharia Sequencial os custos aumentam muito com a proximidade do início da produção e depois caem durante a fase de lançamento, e algumas semanas depois, tornam a subir criando um segundo repique nos custos. Esta nova elevação da curva acontece devido ao fato dos engenheiros percebem que precisam fazer mudanças para atender as necessidades de fabricação. Praticamente o projeto já se encontra na fase final e, então, o custo para recuperar o projeto é maior. Já na Engenharia Simultânea acontece o contrário: os gastos maiores acontecem antes do lançamento, ainda na fase de planejamento, diminuindo e desaparecendo no início da produção.

A análise de Hartley das diferenças dos custos acarretados pelas Engenharias Sequencial e Simultânea durante as fases do projeto está diretamente relacionada ao esforço no projeto, que por sua vez esta diretamente associada à produtividade. Este trabalho limita-se a produtividade e não contempla análise de custos. Esta menção ao trabalho de Hartley é devido ao fato de Prasad (1996) ter se baseado nele em seu trabalho e será mencionado após a figura 1.

A figura 1 ilustra as diferenças dos custos acarretados pelas Engenharias Sequencial e Simultânea durante as fases do projeto.

Figura 1 – Comparação entre os custos das Engenharias Sequencial e Simultânea

Fonte: Hartley, 1992

Segundo Prasad (1996), ao implantar a engenharia simultânea, há uma significativa melhora da qualidade e redução no tempo de desenvolvimento, mudanças de projeto, refugos, retrabalho, defeitos, tempo de introdução do produto e frequência de falha de campo. A tabela 1 mostra, em percentuais, os ganhos obtidos pelas empresas com a aplicação da Engenharia Simultânea ao invés da Sequencial.

Tabela 1 – Percentuais Obtidos com a Implantação da Engenharia Simultânea.

tempo de desenvolvimento	30 – 50% menor
mudanças de engenharia	60 – 95% menor
refugos e retrabalhos	75% de redução
defeitos	30 – 85% menor
tempo de introdução do produto	20 – 90% menor
frequência de falha de campo	60% menor
qualidade em geral	100 – 600% maior

Fonte: Prasad (1996)

Loureiro (1999) desenvolveu um *framework* e um método que integram engenharia de sistemas e engenharia simultânea. O laboratório de engenharia de sistemas do Instituto Tecnológico de Aeronáutica (ITA) implementa esse método

usando uma ferramenta computacional em ambiente de engenharia de sistemas chamada Cradle (3SL, 2010).

Segundo Loureiro e Pires (2010), as características inerentes à engenharia simultânea são:

- Trabalho simultâneo em diversas áreas, que acarreta uma diminuição geral do tempo de desenvolvimento. Por exemplo, o planejamento de processo pode ser integrado com o planejamento de produção.
- Envolvimento dos participantes no trabalho mais cedo. Isso facilitaria a troca de informações entre os diversos setores do trabalho podendo contribuir para um projeto melhor, devido a diferentes ideias geradas no contato entre projetistas e manufatura.
- Trabalho em equipe. Percebe-se que quando as pessoas direcionam-se para um objetivo maior que é realmente o trabalho em equipe, são geradas soluções mais criativas, aumentando o aprendizado entre diversas áreas da empresa.

Embora não há na literatura o uso da engenharia simultânea para resolver o problema de produtividade e aprimorar a qualidade da programação em pares. Engenharia simultânea e programação em pares tem características em comum: trabalho em equipe; cooperação; comunicação; equipes multidisciplinares; melhorar a qualidade; compartilhar o conhecimento; aumentar o aprendizado. Adicionalmente a engenharia simultânea tem duas características complementares à programação em pares: aumentar a produtividade e aprimorar a qualidade.

1.2 Objetivo

O objetivo dessa dissertação é definir, elaborar e implementar uma técnica de desenvolvimento de sistemas que aprimore a melhoria da qualidade, a redução do número de defeitos da programação em pares com objetivo de uni-las ao aumento da produtividade da engenharia simultânea; bem como realizar experimentação prática na utilização desta nova técnica em ambiente controlado e estudos de caso em ambiente corporativo.

1.3 Contribuições

A pesquisa conduzida por Smith et al. (2002) na universidade americana West Virginia e NASA, constatou o problema de produtividade da programação em pares e foi concluído que a NASA não deveria usar método ágil baseado em programação em pares em 64% dos seus projetos. Para os demais casos, o método usado na pesquisa apenas endossou vantagem sobre abordagens convencionais em um relativamente pequeno e específico conjunto de casos, descritos a seguir:

- O projeto é relativamente pequeno;
- Existe uma abundância de desenvolvedores;
- É necessário um período de desenvolvimento curto.

Este trabalho define, elabora e implementa uma nova técnica com intento de aumentar o conjunto de casos constatados por Smith como vantajosos de adotarem um método de desenvolvimento de *software* baseado em programação em pares.

Dyba et al. (2007) constatou aumento do esforço no projeto, a melhoria da qualidade e redução do número de defeitos da programação em pares. Este trabalho contribui ao reorganizar a programação em pares, baseando-se no trabalho de Loureiro et al. (2010) de integração da engenharia de sistemas e engenharia simultânea, visando diminuição do esforço no projeto e aprimoramento da qualidade.

Wray (2010) sugeriu refazer a pesquisa do Dyba (2007) considerando seus quatro mecanismos para melhorar o desempenho da programação em pares, pensar em novos critérios e método para mensurar eficientemente o desempenho levando em consideração estes mecanismos. Este trabalho refaz a pesquisa de Dyba et al. (2007) usando os mecanismos de Wray (2010) e incluindo uma nova técnica baseada na fusão da programação em pares e engenharia simultânea.

1.4 Método de trabalho

Visando o cumprimento do objetivo, o método adotado é o de pesquisa bibliográfica e de experimentação prática na criação, elaboração e utilização de

uma nova técnica de desenvolvimento de *software*, incluindo experimentos em ambiente controlado e estudos de caso em ambiente corporativo.

A pesquisa bibliográfica contempla programação em pares, métodos e critérios para mensurar a produtividade, qualidade e redução do número de defeitos em técnicas de desenvolvimento de *software* em pares. A pesquisa sobre engenharia simultânea abrange desde seu histórico até adequações para sistemas parar servir de base para a fusão proposta por esta dissertação.

O trabalho é dividido nas seguintes **atividades**:

a) Atividade: Revisão da bibliografia.

- **Finalidade:** Trata-se do levantamento dos conceitos de engenharia simultânea e sua aplicação na área de sistemas; conceitos e definição de programação em pares usados por esta dissertação; pesquisa de métodos e critérios para avaliar a produtividade de técnicas de desenvolvimento de *software* em pares, para servir de base para este trabalho.

b) Atividade: Definição, descrição e especificação da nova técnica de desenvolvimento de sistemas.

- **Finalidade:** Definir a nova técnica e principal objeto de pesquisa deste trabalho, para servir de base para descrição, especificação e permitir avaliação e comparação com outras técnicas; descrever como trabalhar em pares usando uma equipe multidisciplinar; especificar a responsabilidade e autoridade dos pares para as decisões, bem como o paralelismo durante a execução da tarefa.

c) Atividade: Definição das medidas e métricas para avaliar a nova técnica

- **Finalidade:** Definir as medidas e métricas para permitir a realização de uma avaliação dos resultados acerca da produtividade, qualidade e do número de defeitos, usando como referência a programação em pares.

Serão usadas várias medidas e métricas nos experimentos e estudos de caso, por exemplo:

- **Total de Linhas de Código (TLOC):** representa o número total de linhas de código de produção do sistema, descartando linhas em branco e comentários. Outra forma comum de utilização é contar aos milhares (*Thousand Lines of Executable Code* ou KLOEC).
- **Número de Pontos(*story-points*) ou “horas ideais” Entregues:** representa o número total de pontos implementados e aceitos pelo cliente. Beck e Fowler (2000) sugerem a utilização de “horas ideais” nas estimativas e controle da iteração, mas a unidade de medida efetivamente utilizada não importa tanto, contanto que seja usada consistentemente durante o projeto.
- **Desvio do padrão de codificação:** representa o percentual total de linhas de código não aderentes ao padrão de codificação.
- **Número de defeitos:** representa o número total de defeitos encontrados, é usado em conjunto com KLOEC e *story-points*. Se aumentar o número de defeitos, é um indicador de que a qualidade do software piorou.
- **Estimativas Originais:** representa o total de pontos (ou horas) originalmente estimadas pela equipe para todas as histórias da iteração.
- **Estimativas Finais:** representa o total de pontos (ou horas, ou “horas ideais”) efetivamente reportadas como gastas para implementar as histórias da iteração.
- **Número de Estórias Entregues:** representa o número total de estórias implementadas e aceitas pelo cliente.

d) Atividade: Aplicação da nova técnica em ambiente controlado

- **Finalidade:** Realizar experimentos para mensurar a produtividade e qualidade da nova técnica em comparação à programação em pares, em ambiente controlado, dentro de empresas e eventos de engenharia de *software*, que permitam mensurar, classificar e relatar a experiência com programação dos participantes, produtividade obtida durante o desenvolvimento de *software* proposto pelo experimento e qualidade do *software* desenvolvido durante o experimento.

e) Atividade: Aplicação da nova técnica em ambiente corporativo

- **Finalidade:** Realizar estudos de caso em empresas de desenvolvimento de *software*, faculdades e corretoras de valores que permitam mensurar, classificar e relatar a experiência com programação dos participantes, produtividade obtida durante o desenvolvimento de *software* no estudo

de caso, qualidade e o número de defeitos em comparação à programação em pares.

f) Atividade: Avaliação dos Resultados

- **Finalidade:** Avaliar a eficácia da nova técnica usando como referencia a programação em pares, verificando nos experimentos e estudos de caso se a nova técnica foi mais produtiva, usando como métrica linhas de código e *story-points* por programador, se preservou a qualidade usando como métrica desvio do padrão de codificação e o número de defeitos.

g) Atividade: Conclusão e Recomendações de trabalhos futuros

- **Finalidade:** Encerra-se a pesquisa com a explanação das suas limitações, e recomendações para trabalhos futuros.

1.5 Organização do trabalho

A seção 2, Programação em pares e engenharia simultânea, apresenta os conceitos básicos de engenharia simultânea, programação em pares, pesquisas sobre produtividade da programação em pares e engenharia simultânea aplicada a sistemas.

A seção 3, Medidas e métricas de avaliação da nova técnica, apresenta um método com critérios já conhecidos, publicados e aceitos, visando permitir a realização de uma avaliação com resultados mensuráveis acerca da produtividade, qualidade e redução do número de defeitos, usando como referência a conhecida e amplamente estudada técnica de programação em pares.

A seção 4, Técnica baseada na fusão da programação em pares e engenharia simultânea, detalha a nova técnica e principal objeto de pesquisa deste trabalho, para servir de base para descrição, especificação e permitir avaliação e comparação com outras técnicas; descreve como trabalhar em pares usando uma equipe multidisciplinar; especifica a responsabilidade e autoridade dos pares para as decisões, bem como o paralelismo durante a execução da tarefa.

A seção 5, Experimentos, apresenta o sumário dos experimentos realizados, em ambiente controlado, concentrando-se na produtividade obtida durante o desenvolvimento de *software* proposto pelo experimento, qualidade e o número de defeitos em comparação à programação em pares.

A seção 6, Estudos de caso, apresenta o sumário de todos estudos de caso realizados em empresas de desenvolvimento de *software*, faculdades e corretoras de valores, concentrando-se na produtividade obtida durante o desenvolvimento de *software* proposto pelo experimento, qualidade e o número de defeitos em comparação à programação em pares.

A seção 7, Análise dos resultados, detalha e analisa os resultados obtidos durante a pesquisa, culminando com a avaliação da nova técnica proposta, baseando-se nos resultados para avaliar sua eficácia.

A seção 8, Conclusão e trabalhos futuros, apresenta a avaliação final da pesquisa, discorrendo sobre as contribuições obtidas e sugestões para trabalhos futuros.

O apêndice A, Detalhamento dos experimentos, apresenta em detalhes todos os experimentos realizados.

O apêndice B, Detalhamento dos estudos de caso, apresenta em detalhes todos os estudos de caso realizados.

1.6 Cronograma

O cronograma está dividido em dois. O primeiro é anterior à matrícula do autor no curso de mestrado. Nesta fase o autor teve a idéia da nova técnica e resolveu realizar experimentação prática em seu trabalho. O resultado foi documentado pelo autor, publicado e apresentado no dia 04 de outubro de 2013 no evento Agile Trends (KATTAN, 2013). Primeiro cronograma:

	04/10	05/10	07/10	07/11	08/11	09/11	05/14
Definição do objeto da pesquisa							
Definição do método e critérios para avaliar a nova técnica							
Aplicação da nova técnica em ambiente corporativo							
Avaliação dos Resultados							
Revisão da bibliografia							

O segundo cronograma é referente ao período no qual o autor já estava cursando o mestrado no IPT:

	05/13	09/13	01/14	04/14	07/14	10/14	12/14
Definição da nova técnica de desenvolvimento de <i>software</i>							
Descrição dos passos e especificação da nova técnica							
Aplicação da nova técnica em ambiente controlado							
Aplicação da nova técnica em ambiente corporativo							
Avaliação dos Resultados							
Conclusão e Recomendações de trabalhos futuros							
Qualificação							
Defesa (prazo limite do contrato sem trancamento: 09/15)							

2 PROGRAMAÇÃO EM PARES E ENGENHARIA SIMULTÂNEA

2.1 – Introdução

Esta seção apresenta as definições e fundamentos da **programação em pares** e **engenharia simultânea**. A Engenharia Simultânea é também conhecida por outros nomes, tais como Engenharia Concorrente, Projeto Concorrente e Desenvolvimento de Produtos Integrados. Nesta dissertação é empregado o termo **Engenharia Simultânea**. Após a apresentação das definições, é apresentado um quadro contendo as características em comum e complementares da programação em pares e engenharia simultânea.

2.2 – Programação em Pares

O código produzido usando o método XP (*Extreme Programming*) é escrito por pares de programadores, que possuem papéis distintos, sentados lado a lado e usando o mesmo computador. Um parceiro é responsável pela codificação e pensa nos algoritmos e na lógica de programação. O outro parceiro observa o código produzido e tenta pensar mais estrategicamente em como melhorá-lo e torná-lo mais simples, além de verificar possíveis erros e pontos de falha.

Além disso, as duplas são constantemente trocadas e os papéis também, objetivando permitir a todos os membros da equipe ter conhecimento sobre todas as partes do sistema, resolvendo o problema de um único desenvolvedor deter o conhecimento sobre uma parte do sistema. Disseminando o conhecimento, mais pessoas participam do desenvolvimento de uma funcionalidade e aprendem como o sistema ou uma ferramenta de desenvolvimento funciona.

A inclusão de um novo membro na equipe que não conhece o sistema cria o problema do primeiro contato com o código. A programação em pares contribui para resolução deste problema, pois o novo membro da equipe pode trabalhar em par com alguém mais experiente. Outra vantagem é propiciar a um programador menos

qualificado tecnicamente aprender com os melhores, aumentando o nível técnico da equipe.

A definição da programação em pares usada nesta dissertação é: dois programadores trabalham colaborativamente na mesma atividade, sentados lado a lado em frente a um único computador. Enquanto uma pessoa está escrevendo o código, por exemplo, a outra observa atentamente o trabalho produzido, buscando defeitos e sugestões de melhoria. A programação em pares estimula a disseminação do conhecimento, reduz a quantidade de defeitos e gera *software* com mais qualidade. (BEGEL; NAGAPPAN, 2008)

A pesquisa conduzida por Smith et al. (2002) na universidade americana West Virginia e NASA, reconheceu o papel fundamental de uma das práticas: a programação em pares. Ela relaciona-se diretamente a quase todas as outras práticas do método XP. A centralidade da programação em pares para o método XP é evidenciada devido ao fato de que caso removida afetaria diretamente todas as práticas relacionadas. Constatou-se o problema de produtividade da programação em pares e foi concluído que a NASA não deveria usar método ágil baseado em programação em pares em 64% dos seus projetos. Para os demais casos, o método usado na pesquisa apenas endossou vantagem sobre abordagens convencionais em um relativamente pequeno e específico conjunto de casos, descritos a seguir:

- O projeto é relativamente pequeno;
- Existe uma abundância de desenvolvedores;
- É necessário um período de desenvolvimento curto.

Segundo Dyba et al. (2007) não é sempre que ocorre a melhoria da qualidade ao usar a programação em pares, conforme é descrito no Quadro 1. Quanto mais experiente é o programador e menos complexa for a tarefa, menor será a melhoria da qualidade.

Quadro 1 – Diretriz para quando usar a programação em pares

Experiência do programador	Complexidade da tarefa	É vantajoso usar a programação em pares?
Júnior	Simplex	Sim, se o aumento da qualidade é o principal objetivo.
Júnior	Complexa	Sim, se o aumento da qualidade é o principal objetivo.
Pleno	Simplex	Não.
Pleno	Complexa	Sim, se o aumento da qualidade é o principal objetivo.
Sênior	Simplex	Não.
Sênior	Complexa	Não, a menos que a tarefa é complexa demais para ser resolvida por apenas um programador sênior.

Fonte: Dyba et. al., 2007, p. 15

Legenda: Júnior = programador iniciante; Pleno = programador de experiência intermediária; Sênior = programador experiente.

Cockburn et al. (2001) e Williams et al. (2000) usaram programadores iniciantes e constataram que a produtividade ao usar a programação em pares parece ser comparável à de duas pessoas trabalhando de forma independente. As razões sugeridas são que programando em pares discutem o sistema antes de desenvolvê-lo, então provavelmente ocorrerá menos erros e retrabalho. Além disso, o número de erros evitados pela inspeção informal realizada pela programação em pares é tal que, menos tempo é gasto na reparação de *bugs* descobertos durante o processo de testes.

Segundo Parrish et al. (2004), há aumento de produtividade quando o desenvolvimento é realizado de maneira concorrente por meio da programação individual, quando os programadores trabalham em computadores e partes diferentes do sistema. Porém a qualidade demonstrou-se melhor quando usam a programação em pares. A produtividade foi mensurada pelo número de pontos de

função desenvolvido por hora por programador. As equipes foram classificadas em níveis de concorrência(simultaneidade) alto e baixo. A qualidade foi mensurada pelo desvio do padrão de codificação. As equipes estavam no mesmo local físico.

Arisholm et al. (2007) e Parrish et al. (2004) realizaram seus trabalhos com programadores experientes. Eles descobriram que houve uma significativa perda de produtividade em comparação a dois programadores trabalhando sozinhos, havia alguns benefícios de qualidade, mas estes não compensavam totalmente a perda da produtividade da programação em pares. No entanto, segundo eles, a partilha de conhecimento que acontece durante a programação em pares é muito importante, pois reduz os riscos globais para um projeto no caso de algum membro da equipe sair e isso pode justificar o seu uso.

O estudo liderado por Dyba et al. (2007) revisou quinze estudos sobre programação em pares, quatro foram executados com profissionais e onze com alunos. Dez pesquisas eram da Europa e cinco da América do Norte. Onze compararam a efetividade de pares isolados contra indivíduos isolados, e quatro estudos fizeram a comparação em um contexto de equipe (isto é, num contexto em que múltiplos indivíduos ou pares trabalharam juntos para desenvolver um produto de *software*). Todos os estudos envolveram tarefas de programação como a base para a comparação. O número de indivíduos nos estudos oscilou entre 12 e 295. O estudo concentrou-se nos aspectos:

- Duração, tempo do calendário de desenvolvimento.
- Esforço, mensurado em horas de trabalho.
- Qualidade, quanto melhor ficou o produto final.

Ao analisar-se o resultado, constata-se a melhora da qualidade, redução do número de defeitos, porém observa-se também ao usar a programação em pares o aumento do esforço no projeto, se comparada à programação individual.

Wray (2010) apresentou quatro mecanismos para melhorar o desempenho da programação em pares:

- **Comunicação.** Parte da eficácia da programação em pares é presumivelmente devido à interação entre os programadores. Quando os programadores falam sobre o problema no momento da dúvida, encontram mais rapidamente a solução, consequentemente há o aumento da produtividade;
- **Rotação do par.** A fadiga diminui a produtividade e trocar a pessoa do par que digita no teclado durante o expediente, pode aumentar a produtividade. É importante levar em consideração que programadores iniciantes aumentam sua produtividade quando seu par é mais experiente.
- **Lutar contra práticas ruins de programação.** A pressão do par contribui para melhorar a qualidade da programação, evitando retrabalhos e aumentando a produtividade;
- **Compartilhar e reconhecer competências.** A descoberta de quem são os especialistas em determinados assuntos e compartilhar o conhecimento, pode aumentar a produtividade.

Wray (2010) sugeriu refazer a pesquisa do Dyba (2007) considerando seus quatro mecanismos e pensar em novos critérios e método para mensurar eficientemente o desempenho, levando em consideração estes mecanismos.

Segundo Sommerville (2011) a utilização da técnica de programação em pares tem uma série de vantagens:

- Ela suporta a idéia de propriedade e responsabilidade coletiva para o sistema. Isso reflete a idéia de Weinberg (1971) de programação sem ego, onde o *software* é de propriedade da equipe como um todo e os indivíduos não são responsáveis pelos problemas com o código. Em vez disso, a equipe tem responsabilidade de resolver estes problemas;
- Ela atua como um processo de revisão informal, porque cada linha de código é olhado por pelo menos duas pessoas. Inspeções de código ajudam muito na descoberta de uma elevada percentagem de erros de *software*. No entanto, são demorados para organizar e, geralmente, representam atrasos no processo de desenvolvimento. Embora a

programação em pares é um processo menos formal e, provavelmente, não encontra tantos erros como inspeções de código, é muito mais barato que um processo de inspeção formal;

- Ela apoia a refatoração, que é um processo de melhoria de *software*. A dificuldade de implementar isso em um ambiente normal de desenvolvimento de *software* é que o esforço em refatoração é gasto no presente para o benefício a longo prazo no futuro. Um indivíduo que pratica a refatoração pode ser considerado menos eficiente do que aquele que simplesmente carrega os problemas no desenvolvimento do código. Quando são utilizados a programação em pares e propriedade coletiva, outros se beneficiam imediatamente da refatoração e eles tendem a apoiar o processo, pois reconhecem seu benefício.

Segundo Sillitti (2012), ao analisar-se o resultado, as pessoas mudam de forma relevante seus hábitos de trabalho quando trabalham em pares e tendem a concentrar-se mais em atividades produtivas. As sessões de trabalho são significativamente mais longas e ininterruptas e eles concentram-se mais nas tarefas, diminuindo distrações como e-mails, navegadores de internet e programas de bate-papo. Ao mesmo tempo é encorajada a discussão, melhorando a solução criada e ajudando o par a ter uma visão mais clara do problema a ser resolvido.

Bella (2013) publicou um estudo de caso que foi realizado para avaliar o efeito da programação em pares na qualidade e eficiência das correções de defeitos. Foi estudado um projeto de desenvolvimento de *software* em uma grande empresa italiana, por um prazo de 14 meses. Comparado com estudos de casos existentes de programação em pares, o período mais longo torna o estudo mais realista. Em análise exploratória, a efetividade da programação em pares foi investigada no contexto de correções de defeitos e implementações de histórias de usuários. A análise mostra que a introdução de novos defeitos tende a diminuir quando a programação em pares é usada. Os resultados são consistentes para ambos os contextos.

Foi realizado por Plaue (2013) um estudo experimental de programação em pares contra programação individual. O resultado mostra que a programação em pares é uma intervenção de baixo custo para retenção de pessoas na equipe.

A base do conceito da **Engenharia Simultânea** é melhorar a qualidade e aumentar a produtividade com redução de custos. Ainda não há na literatura o uso deste conceito para aumentar a produtividade e aprimorar a qualidade da programação em pares.

2.3 – Engenharia Simultânea

O conceito de **Engenharia Simultânea** é baseado na ideia do processamento paralelo/simultâneo dos processos da empresa, que reduz o tempo de lançamento de um novo produto bem como a melhoria da qualidade (PITHON, 2004, p. 3).

Em 1982 foi iniciado um estudo, conduzido pelo DARPA (*Defense Advanced Research Project Agency* – Agência de Projetos de Investigação Avançada de Defesa), buscando uma forma de aumentar o grau de paralelismo das atividades de desenvolvimento de produtos. O resultado deste trabalho, publicado em 1988, definiu o termo Engenharia Simultânea da seguinte forma:

Engenharia Simultânea é uma abordagem sistemática para o desenvolvimento integrado e paralelo do projeto de um produto e os processos relacionados, incluindo manufatura e pós-venda. Esta abordagem procura fazer com que as pessoas envolvidas no desenvolvimento considerem, desde o início, todos os custos, prazos e requisitos dos clientes. (Winner et al., 1988 apud (Prasad, 1996))

Nas entrelinhas desta definição, o envolvimento de representantes das várias áreas que agregam conhecimento e experiência ao produto podem e devem ser chamadas a participar. Vem daí a necessidade da formação de equipes interdisciplinares coordenadas e voltadas a um objetivo final: satisfazer as necessidades dos clientes, o que sem dúvida alguma trará retorno financeiro às corporações. Marketing, vendas, assistência técnica, testes, fabricação, engenharia, expedição e demais áreas do conhecimento devem ser envolvidas, se não em dedicação completa, pelo menos parcialmente.

A partir da definição de Winner, o conceito de Engenharia Simultânea tornou-se muito mais abrangente, podendo incluir a cooperação e o consenso entre os envolvidos no desenvolvimento, com o emprego de recursos computacionais. A partir desta definição inicial, muitas outras surgiram:

Engenharia simultânea é uma abordagem sistemática para o desenvolvimento integrado de produtos que enfatiza o atendimento das expectativas dos clientes. Inclui os valores de trabalho em equipe, tais como cooperação, confiança e compartilhamento, de forma que as decisões sejam tomadas, no início do processo, em grandes intervalos de trabalho paralelo incluindo todas as perspectivas do ciclo de vida, sincronizadas com pequenas modificações para produzir consenso (Ashley, 1992 apud Prasad, 1996)

Engenharia simultânea é um ambiente de desenvolvimento, no qual a tecnologia de projeto auxiliado por computador é utilizada para melhorar a qualidade do produto, não somente durante o desenvolvimento, mas em todo o ciclo de vida (Ellis, 1992 apud Prasad, 1996)

Engenharia simultânea é uma metodologia de desenvolvimento de produtos, na qual várias habilidades (*X-abilities*) são consideradas parte do processo de desenvolvimento de produtos (manufatura, serviços, qualidade entre outros). Esses requisitos não servem somente para se atingir as funcionalidades básicas do produto, mas para definir um produto que atenda todas as necessidades dos clientes (Hartley, 1992 apud Prasad, 1996)

A Engenharia simultânea é uma idéia aparentemente simples baseada fundamentalmente nas diferentes formas de como os produtos são concebidos, projetados e produzidos. A ideia é a cooperação mútua entre as pessoas, de modo que a de se obter um melhor desempenho a fim de alcançar com sucesso a meta comum (Causing, 1989)

Todas estas definições abordam de alguma maneira várias palavras consideradas chaves para o sucesso da implementação da Engenharia Simultânea, tais como: trabalho em equipe, cooperação, qualidade do produto e ciclo de vida. Estas definições também partilham a hipótese de que a Engenharia Simultânea é o meio para aprimorar a qualidade do projeto do produto com a redução dos custos.

Segundo Mills e Beckert (1991), a tabela 2, mostra as diferenças entre engenharia tradicional e simultânea, de acordo com o enfoque de mercado, estratégia, qualidade e projeto.

Tabela 2 – Engenharia Tradicional versus Engenharia Simultânea

ENFOQUE	ENGENHARIA TRADICIONAL	ENGENHARIA SIMULTÂNEA
MERCADO	<ul style="list-style-type: none"> ♦ produto desenvolvido por critérios técnicos ♦ o fabricante define o produto ♦ pouco envolvimento de <i>marketing</i> 	<ul style="list-style-type: none"> ♦ produto desenvolvido a partir de critérios do consumidor ♦ o mercado define o produto ♦ grande envolvimento de <i>marketing</i>
ESTRATÉGIA	<ul style="list-style-type: none"> ♦ calcado na visão interna ♦ ênfase no preço ♦ limitado por regras ♦ objetivos econômicos ♦ qualidade do produto 	<ul style="list-style-type: none"> ♦ calcado na visão externa ♦ ênfase no custo ♦ ambiente de alta competição ♦ objetivos econômicos e sociais ♦ qualidade total
QUALIDADE	<ul style="list-style-type: none"> ♦ qualidade focada no controle ♦ pouco uso de ferramentas para garantir a qualidade no projeto ♦ qualidade avaliada (correção) 	<ul style="list-style-type: none"> ♦ qualidade focada no projeto ♦ uso intensivo de ferramentas para assegurar a qualidade do projeto ♦ qualidade projetada (prevenção)
PROJETO	<ul style="list-style-type: none"> ♦ o processo adapta-se ao produto ♦ uso da prancheta ♦ muitas alterações de engenharia ao longo da vida do produto ♦ longo tempo de desenvolvimento ♦ desenvolvido pela engenharia do produto 	<ul style="list-style-type: none"> ♦ produto e processos compatíveis ♦ uso do CAE/CAD/CAM ♦ poucas alterações de engenharia ao longo da vida do produto ♦ redução do tempo de desenvolvimento ♦ desenvolvido pela equipe multidisciplinar

Fonte: Mills, 1991

Segundo Bennett et al. (1995) o principal objetivo da Engenharia Simultânea é a diminuição do tempo desde o pedido até a entrega, para um novo produto, com custo mais baixo e maior qualidade. Isto é alcançado através do desenvolvimento paralelo, ao invés de sequencial, das diferentes etapas que compõem o Projeto do Produto, com o emprego de times ou equipes multidisciplinares. Assim como a filosofia do *Just-in-Time*, a Engenharia Simultânea também é uma filosofia e não uma tecnologia. Engenharia Simultânea usa tecnologia para atingir seus objetivos.

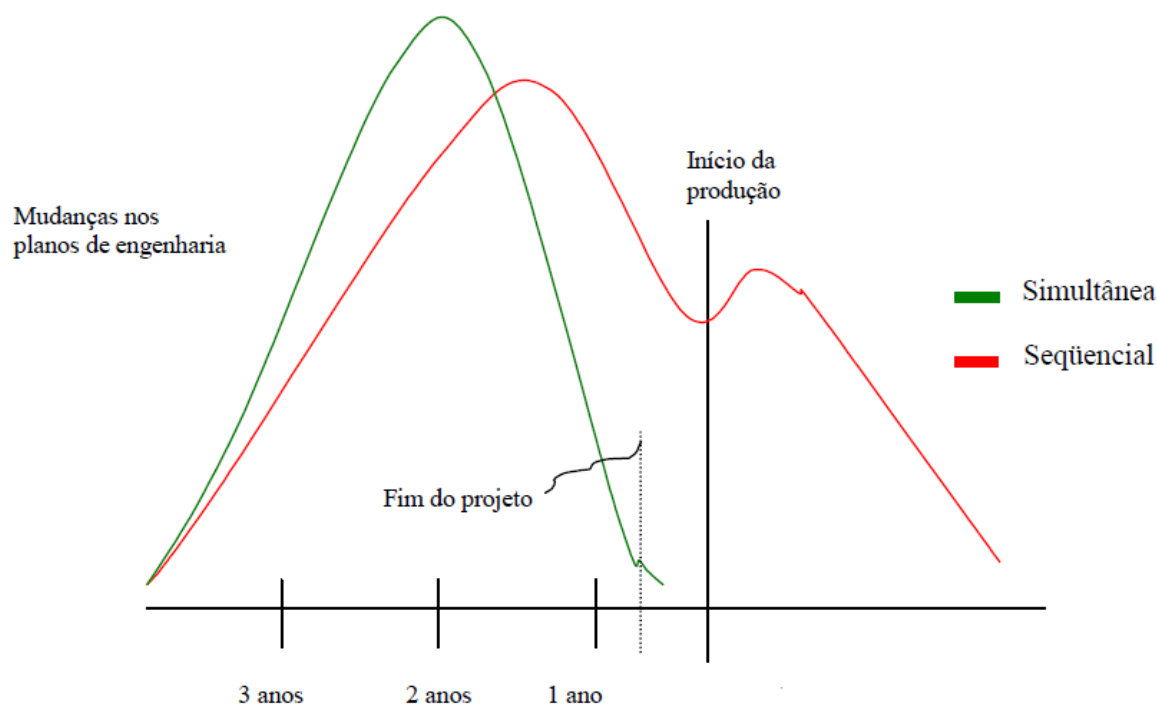
Segundo Hartley (1992) na Engenharia Sequencial os custos aumentam muito com a proximidade do início da produção e depois caem durante a fase de lançamento, e algumas semanas depois, tornam a subir criando um segundo repique nos custos. Esta nova elevação da curva acontece devido ao fato dos engenheiros percebem que precisam fazer mudanças para atender as necessidades de fabricação. Praticamente o projeto já se encontra na fase final e, então, o custo para

recuperar o projeto é maior. Já na Engenharia Simultânea acontece o contrário: os gastos maiores acontecem antes do lançamento, ainda na fase de planejamento, diminuindo e desaparecendo no início da produção.

A análise de Hartley das diferenças dos custos acarretados pelas Engenharias Sequencial e Simultânea durante as fases do projeto está diretamente relacionada ao esforço no projeto, que por sua vez está diretamente associada à produtividade. Este trabalho limita-se a produtividade e não contempla análise de custos. Esta menção ao trabalho de Hartley é devido ao fato de Prassad (1996) ter se baseado nele em seu trabalho e será mencionado após a figura 1.

A figura 1 ilustra as diferenças dos custos acarretados pelas Engenharias Sequencial e Simultânea durante as fases do projeto.

Figura 1 – Comparação entre os custos das Engenharias Sequencial e Simultânea



Fonte: Hartley, 1992

Segundo Prassad (1996), ao implantar a engenharia simultânea, há uma significativa melhora da qualidade e redução no tempo de desenvolvimento, mudanças de projeto, refugos, retrabalho, defeitos, tempo de introdução do produto

e frequência de falha de campo. A tabela 1 mostra, em percentuais, os ganhos obtidos pelas empresas com a aplicação da Engenharia Simultânea ao invés da Sequencial.

Tabela 1 – Percentuais Obtidos com a Implantação da Engenharia Simultânea.

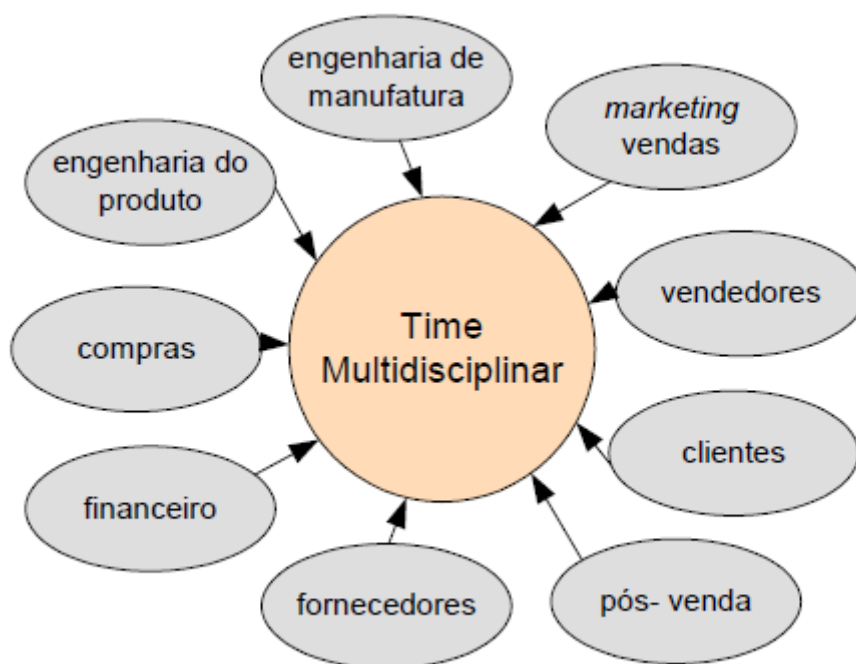
tempo de desenvolvimento	30 – 50% menor
mudanças de engenharia	60 – 95% menor
refugos e retrabalhos	75% de redução
defeitos	30 – 85% menor
tempo de introdução do produto	20 – 90% menor
frequência de falha de campo	60% menor
qualidade em geral	100 – 600% maior

Fonte: Prasad (1996)

Para alcançar as propostas da Engenharia Simultânea, é fundamental a formação de uma equipe multidisciplinar com pessoas de todas as áreas e especialidades envolvidas no projeto. Esta equipe pode crescer ou diminuir ao longo de sua existência, mantendo sempre um mesmo núcleo de pessoas que acompanham o desenvolvimento. A equipe deve trabalhar em sintonia, considerando todos os detalhes, para que o trabalho realizado em cada área disciplinar seja compatível com as demais e que cada área influa na outra com informações corretas e no tempo certo.

O sucesso deste grupo está diretamente relacionado ao engajamento do “topo” da organização, que deve mostrar um total apoio e dispor para a equipe os meios para a realização do trabalho. A alta direção transfere, então, para equipe o dever de levar o projeto adiante, cobrando periodicamente o andamento do projeto. Sendo a força-tarefa composta de vários membros com formação profissional específica e diferenciada, as especificações do produto passam a ter um maior grau de definição, acarretando, deste modo, um menor custo nas possíveis redefinições do produto. A figura 2 mostra os componentes do time Multidisciplinar de desenvolvimento de produto.

Figura 2 – Composição do Time Multidisciplinar de Desenvolvimento de Produtos



Fonte: Hartley, 1992

Uma característica importante deste time de Engenharia Simultânea é ser responsável por todo o projeto e possuir autoridade para as decisões. Esta atitude requer treinamento dos membros do time e da gerência para ser efetiva.

Além disso, para que a Engenharia Simultânea tenha sucesso, é preciso que exista a comunicação efetiva entre os seus integrantes. Esta comunicação envolve as pessoas, a troca de dados entre os sistemas utilizados e, mais ainda a documentação e o gerenciamento das informações e das decisões, para que os dados e a documentação possam ser recuperadas sempre que necessário.

Loureiro (1999) desenvolveu um *framework* e um método que integram engenharia de sistemas e engenharia simultânea. O laboratório de engenharia de sistemas do Instituto Tecnológico de Aeronáutica (ITA) implementa esse método usando uma ferramenta computacional em ambiente de engenharia de sistemas chamada Cradle (3SL, 2010).

Segundo Loureiro e Pires (2010), as características inerentes à engenharia simultânea são:

- Trabalho simultâneo em diversas áreas, que acarreta uma diminuição geral do tempo de desenvolvimento. Por exemplo, o planejamento de processo pode ser integrado com o planejamento de produção.
- Envolvimento dos participantes no trabalho mais cedo. Isso facilitaria a troca de informações entre os diversos setores do trabalho podendo contribuir para um projeto melhor, devido a diferentes ideias geradas no contato entre projetistas e manufatura.
- Trabalho em equipe. Percebe-se que quando as pessoas direcionam-se para um objetivo maior que é realmente o trabalho em equipe, são geradas soluções mais criativas, aumentando o aprendizado entre diversas áreas da empresa.

2.5 Conclusão

Esta seção apresentou os fundamentos, pontos fortes e fracos da programação em pares. Engenharia de simultânea e a relação que existe entre essas duas abordagens de desenvolvimento de *software*. Embora não há na literatura o uso da engenharia simultânea para resolver o problema de produtividade e aprimorar a qualidade da programação em pares. Engenharia simultânea e programação em pares tem características em comum: trabalho em equipe; cooperação; comunicação; equipes multidisciplinares; melhorar a qualidade; compartilhar o conhecimento; aumentar o aprendizado. Adicionalmente a engenharia simultânea tem duas características complementares à programação em pares: aumentar a produtividade e aprimorar a qualidade.

REFERÊNCIAS

3SL, **Cradle manuals**, 3SL, London, 2010

ARISHOLM, E; GALLIS, H.; DYBA, T; SJOBERG, D. **Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise**, IEEE Trans. *Software Eng.*, vol. 33, no. 2, 2007, pp. 65–86.

BECK, K; FOWLER, M. **Planning Extreme Programming**. 1. ed. Boston: Addison-Wesley Professional, 2000.

BEGEL, A; NAGAPPAN, N. **Pair programming: what's in it for me?** In *ESEM '08: Proc. of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, pages 120–128, New York, NY, USA, 2008. ACM.

BELLA, E; FRONZA, I; PHAPHOOM, N; SILLITTI, A; SUCCI, G; VLASENKO, J. **Pair Programming and Software Defects—A Large, Industrial Case Study** IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 39, NO. 7, JULY 2013.

CAUSING, D. **Concurrent Engineering**. American Society of Mechanical Engineers. Cambridge, Mass. 1989.

CHIDAMBER, S; KEMERER, C. **A metrics suite for object oriented design**. IEEE Transactions on Software Engineering, vol. 20, no. 6, p.476–493, 1994

COCKBURN, A.; WILLIAMS, L. **'The costs and benefits of pair programming'**. In *Extreme programming examined.*, Boston, Addison-Wesley, 2001.

DYBA, T.; ARISHOLM, E; SJOBERG, K; HANNAY, E; SHULL, F. **Are Two Heads Better than One? On the Effectiveness of Pair Programming**. IEEE *Software*, vol. 24, no. 6, 2007, pp. 12–15.

HARTLEY, J. R. **Concurrent engineering** : shortening lead times, raising quality, and lowering costs. Cambridge, Mass. : Productivity Press, 1992.

KATTAN, H. M. Programação simultânea em pares. Uma nova técnica ágil: Programação em pares evoluída pela engenharia simultânea. In: AGILE TRENDS, 2013, São Paulo, SP. **Eletronic Proceedings**... Disponível em: <<http://agiletrendsbr.com/2013/programacao/>>. Acesso em: 05 maior 2014.

LOUREIRO, G; PIRES, C.A.P. **Um ambiente computacional para Engenharia de Sistemas** - Anais do XVI Encontro de Iniciação Científica e Pós-Graduação do ITA – XVI ENCITA / 2010 Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil, 20 de outubro de 2010

LOUREIRO, G. **A systems engineering and concurrent engineering framework for the integrated development of complex products**, PhD Thesis, Loughborough University, Loughborough, 1999.

MILLS, R. e BECHERT B. **The Future of Product Development. Computer Aided Engineering**, 1991.

PARRISH, A; SMITH, R; HALE, D; HALE, J. **A Field Study of Developer Pairs: Productivity Impacts and Implications.** *University of Alabama - USA - IEEE Computer Society-*, pp.76-79. Outubro de 2004

PITHON, A. **Projeto Organizacional para a Engenharia Concorrente no Âmbito das Empresas Virtuais.** 2004. 316 f. Tese (Doutorado) - Escola de Engenharia da Universidade do Minho Departamento de Produção e Sistemas. Universidade do Minho, Guinardes, 2004.

PLAUE, C; LI,Z; KRAEMER, E. **A Spirit of Camaraderie: The Impact of Pair Programming on Retention** *Department of Computer Science The University of Georgia Athens, GA USA - IEEE - CSEE&T 2013, San Francisco, CA, USA*, pp.209-218.

PRASAD, B. **Concurrent engineering fundamentals: integrated product development. v. 2.** ed.1. 1996. New Jersey: Prentice Hall. p.326.

SILLITTI, A;SUCCI, G;VLASENKO,J. **Understanding the Impact of Pair Programming on Developers Attention—A Large, Industrial Case Study** *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, JUNE 2012, p.1094-1101

SMITH, J. **Should NASA Embrace Agile Processes.** 27th Annual IEEE/NASA Software Engineering Workshop, Greenbelt, MD, USA, December 5-6, 2002, Greenbelt Marriott.

SYAN, C. S. Introduction to concurrent engineering. In: SYAN, C. S.; MENON, U. **Concurrent engineering: concepts, implementation and practice.** London, England: Chapman & Hall, 1994. Cap.1, p.3-23.

VALLE, S; BUSTELO, D.V. et al, **Concurrent engineering performance: Incremental versus radical innovation.** IN: *Internal Journal Production Economics* , University of Oviedo, 2009

WEINBERG, G. **The Psychology of Computer Programming.** New York: Van Nostrand, 1971.

WILLIAMS, L.; KESSLER, R.; CUNNINGHAM, W.; JEFFRIES, R. **‘Strengthening the Case for Pair Programming’.** *IEEE Software*, 2000, p.19–25.

WINNER, R. I. et al. **The role of concurrent engineering in weapons system acquisition.** E.U.A: Institute for Defense Analyses, 1988. IDA Report R-338.

WRAY, S. **How Pair Programming Really Works.** *IEEE Software*, vol. 24, no.6, 2010, pp. 50–55.

WRAY, S. **Responses to “How Pair Programming Really Works”.** *IEEE Software*, March/April 2010, pp. 8–9.

REFERENCIAS CONSULTADAS

BECK, K. **Extreme Programming Explained: Embrace Change**. 1. ed. Boston: Addison-Wesley, 2000. 58p.

BECK, K. **Extreme Programming Explained: Embrace Change**. 2. ed. Boston: Addison-Wesley, 2004. 75p.

BENNETT, J. G.; LAMB, T. Concurrent engineering : application and implementation for U.S. shipbuilding. In: SHIP PRODUCTION SYMPOSIUM, Seattle, 1995. **Proceedings**. Jersey City, NJ. : SNAME, 1995. p.23.1-23.4.

CHIUSOLI, R. F. (1996). **Engenharia Simultânea: estudos de casos na indústria brasileira de autopeças**. São Carlos. 147p. Dissertação (Mestrado) - Universidade Federal de São Carlos.

LOUREIRO, G. et al. *Systems concurrent engineering to develop a greenn car*. IN: **Proceedings of the CE2010 17th ISPE International Society for Productivity Enhancement**, Springer-Verlag, Cracóvia, Polônia, 2010.

MUNIZ JUNIOR, J. (1995). **Utilizacao da engenharia simultanea no aprimoramento continuo e competitivo das organizacoes: estudo de caso do modelo usado no aviao emb 145 da embraer**. São Paulo. 196p. Dissertação (Mestrado) - Escola Politécnica - Universidade de São Paulo. (Disponível na biblioteca da EP - USP).

PITHON, A. e PUTNIK G. (2001). **Engenharia Concorrente e Times de Trabalho - Introdução**; Relatório Técnico, CESP-GIS-01-01, Universidade do Minho. Guimarães.

PITHON, A. e PUTNIK G. (2002). **Team Work for Concurrent Engineering in Agile/Virtual Enterprise by BM_Virtual Enterprise Architecture Reference Model. Collaborative Business Ecosystems and Virtual Enterprises** - Third Working Conference on Infrastructure for Virtual Enterprises (PRO-VE'02). L. M. Camarinha-Matos. 1-3 Maio, Sesimbra - Portugal, Kluwer Academic Publishers. **52**: 489-496.

PITHON, A. e PUTNIK G. BM_VVEARM como Suporte para a Universidade do Futuro. World Congress on Engineering and Technology Education – Guarujá/Santos, Brasil - WCETE'04 14-17 Março de 2004.

SALLEH, N; MENDES, E; GRUNDY, J., **"Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review,"** IEEE Trans. Softw. Eng., vol. 37, pp. 509-525, jul 2011.

SYAN, C. S.; MENON, U. **Concurrent engineering: concepts, implementation and practice**. 1.ed. London: Chapman & Hall, 1994.

WILLIAMS, L; KESSLER,R. **Pair Programming Illuminated**. 1. ed. Boston: Addison-Wesley, 2003.

WILLIAMS, L; KESSLER,R **The Effects of 'Pair-Pressure' and 'Pair-Learning' on Software Engineering Education**. In: CONF. SOFTWARE ENG. EDUCATION (CSEE&T 00), IEEE CS Press, 2000, pp. 59–65.

WINNER, R. **Integrated product/process development in the new attack submarine program: a case study**. 2.ed. s.l. . : Undersecretary of Defense, 2000.

WINNER, R. **The VIRGINIA class submarine program: a case study**. Groton, CT : General Dynamics Electric Boat, 2002.

YAN, H. S. (1999). "**Agile concurrent engineering**." Integrated Manufacturing