

APLICAÇÃO DE METODOLOGIA LEAN À ENGENHARIA DE SOFTWARE EM UMA EMPRESA DE MÉDIO PORTE

Dionisio Raony de Souza Ribeiro dionisio.raony@gmail.com

Debora Lucia Marcos deboramarcos.94@gmail.com

Eloan Junior Rizelo eloan_jr@hotmail.com

Luana Larissa Moreira Gaiovicz llmgaiovicz@gmail.com

Resumo:

O presente estudo foi realizado em uma empresa de médio porte desenvolvedora de software de automação comercial, localizada na cidade de Dois Vizinhos, Paraná, e teve como principal objetivo analisar o seu processo produtivo, bem como expor a realidade atual da empresa, tendo como foco evidenciar o custo e o tempo empregados para a correção do desperdício encontrado em seu processo produtivo, o retrabalho. Além de desenvolver uma proposta de melhoria para a redução do retrabalho no desenvolvimento do produto, com base na filosofia *Lean* de Software, criada a partir da filosofia *Lean*, sendo essa uma filosofia de gestão inspirada em práticas e resultados do Sistema Toyota de Produção. Estimou-se que aproximadamente 40% do tempo útil de trabalho na empresa em questão é empregado em retrabalho. Levantando a necessidade de verificação do custo relacionado a este desperdício, o qual resultou em R\$ 21.531,08. Diante dos números obtidos, percebe-se que o retrabalho é um desperdício no processo produtivo e deve ser estudado de forma a ser reduzido, para que o colaborador desempenhe suas funções em atividades que agreguem valor ao produto. A Engenharia de Produção tem o *Kaizen* – prática da melhoria contínua, como fator motivador em sua atuação, portanto, esse profissional agrega valor em qualquer processo produtivo, desde a fabricação do produto acabado tangível até a prestação de serviço, que é um produto intangível, visando a eliminação de desperdícios, diminuição de custos e maximização de lucro.

Palavras chave: Engenharia de Produção, Engenharia de Software, retrabalho, otimização de processos.

LEAN METHOD APPLICATION ON SOFTWARE ENGINEERING IN A MIDSIZE COMPANY

Abstract

The present study was realized on a midsize company software developer of commercial automation, localized in the city of Dois Vizinhos, Paraná, and had as main objective to analyze their productive process, as well to expose the current reality of the company, focusing on to evidence the cost and time used to correction of found waste on the productive process, the rework. Besides, to develop an improvement proposal for reduction of rework at product development, with base on Software Lean philosophy, created from Lean philosophy, being this a management philosophy inspired in practices and results of Toyota Production System. It was estimated about 40% of work useful time at company in question is appointed in rework. Bringing up the necessity to verification of cost related with this waste, which results in R\$ 21.531,08. In front of the numbers obtained, it is perceived that the rework is a waste in the productive process and should be studied in a way to be reduced, so that the employee performs his functions in activities that add value to the product. The Production Engineering has the *Kaizen* – practice of continuous improvement, as a motivator factor in your actuation, therefore, this professional adds value at any productive process, since the tangible finished product manufacturing until the provision of services, that is a intangible product, aiming the wastes elimination, cost elimination and profit maximization.

Key-words: Production Engineering, Software Engineering, rework, process optimization

1. Introdução

O mercado brasileiro de Tecnologia da Informação (hardware, software, serviços e exportações de TI) movimentou 39,6 bilhões de dólares em 2016, o que representou 2,1% do PIB brasileiro, de acordo com o estudo Mercado Brasileiro de Software e Serviços 2017, produzido pela Associação Brasileira das Empresas de Software (ABES). No ranking de investimentos na América Latina, o Brasil está em 1º lugar, com 36,5% dos investimentos (ABES, 2017).

Outro fator a se destacar no avanço tecnológico é a necessidade da migração do gerenciamento de suas atividades de forma analógica (papeis, formulários, entre outros) para sistemas digitais de gerenciamento e controle das informações. Além das exigências fiscais determinadas pelo Governo, as quais requerem do empresário o uso de um software homologado para cumprimento destas legislações.

Diante do atual cenário tecnológico, as empresas de software buscam gradativamente manter-se competitivas e atuantes no mercado, dispondo um produto prático e confiável, com custo justo e entregue no tempo certo. Quando se trata da prestação de serviços é de suma importância que o mesmo seja desempenhado de forma a satisfazer totalmente as necessidades do cliente.

No entanto, um dos problemas encontrados no processo produtivo de uma fábrica de software é o retrabalho, e de acordo com Boehm e Basili (2001), representa de 40% a 50% do esforço de um projeto de desenvolvimento de software, os quais poderiam ser evitados.

Portanto faz-se necessário estudos neste setor, evidenciando como a engenharia de produção pode auxiliar o gestor de uma empresa de software a entregar seus produtos com maior qualidade, tornando o seu processo mais eficiente, com menos desperdícios e mais resultados, além de almejar sempre a melhoria contínua, atuando mais competitivo no mercado.

Desta forma, escolheu-se uma empresa de software para aplicação de ferramentas de engenharia de produção com a finalidade de aumento da eficiência no processo produtivo. Com isso, o presente estudo foi realizado em uma Fábrica de Software localizada na cidade de Dois Vizinhos, Sudoeste do Paraná, a qual atende clientes de todo o território nacional e atua há 23 anos no mercado. A empresa em questão desenvolve *software* de Automação Comercial para empresas de diversos ramos de atividade.

Um dos desperdícios da produção identificado em seu processo produtivo é o retrabalho. A partir disso, o estudo tem como principal objetivo apresentar uma proposta para a redução deste desperdício, com base no ciclo PDCA e nos princípios da filosofia *lean* e *lean* de software.

2. Justificativa.

Segundo Kalinowski e Spinola (2008), o esforço empregado com retrabalho em empresas de software, varia em média de 40% a 50% do esforço total do desenvolvimento de um projeto. Inicialmente, na empresa em questão aproximadamente 11,73% dos chamados resultam em retrabalho. Estimou-se que aproximadamente 40% do tempo útil de trabalho na empresa em questão é empregado em retrabalho. Levantando a necessidade de verificação do custo relacionado a este desperdício.

Os desperdícios de produção relacionado ao retrabalho na programação do software, ocasionam demora na entrega do projeto, insatisfação do cliente (interno e externo), atraso no cronograma, custos de mão-de-obra e acúmulo de tarefas.

Gerando ônus de tempo e recursos no processo produtivo. Diante disso, apresenta-se a necessidade de análise e reconfiguração do processo produtivo. Para possibilitar a empresa a

entregar seu produto no tempo certo, com um custo justo, sem sobrecarga de seus colaboradores. Por fim, satisfazendo as necessidades dos clientes e agregando valor ao produto.

3. Metodologia

O presente estudo caracteriza-se como tipo de pesquisa: exploratória (através de levantamento bibliográfico referente ao tema estudado) e descritiva (pois houve observação, registro e análise do fenômeno estudado). Com base nos objetivos a pesquisa exploratória segundo Gil (2002), proporciona maior familiaridade com o problema estudado, e a pesquisa descritiva descreve as características de determinada população ou fenômeno e estabelece relação entre as variáveis analisadas. Com base nos procedimentos técnicos utilizados, o estudo realizou-se através de: pesquisa bibliográfica em livros, artigos, sites, os quais fundamentaram o assunto estudado, como as ferramentas da qualidade, *lean* e engenharia de *software*; e pesquisa documental em relatórios e documentos particulares da empresa, para analisar o cenário atual do problema estudado

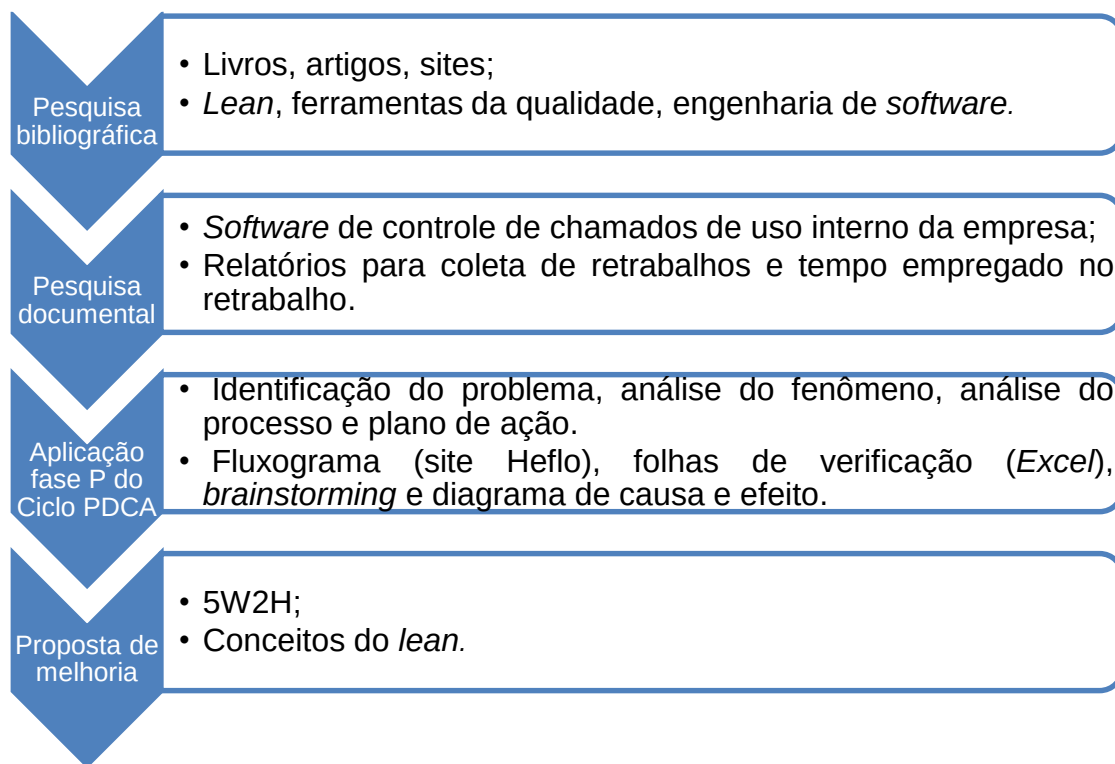


Figura 1 – Etapas do trabalho realizado

A Figura 1 representa as etapas realizadas na construção do presente estudo, as quais se deram através de pesquisa bibliográfica do assunto abordado, pesquisa documental em *software* da empresa para coleta de retrabalhos e tempo empregado em retrabalhos. Aplicação da etapa P do Ciclo PDCA, na qual utilizaram-se as ferramentas: *Excel* para gerar as folhas de verificação dos retrabalhos e tempos de retrabalho, site Heflo para desenvolver o fluxograma do ciclo do chamado, *brainstorming* e diagrama de causa e efeito. E por fim, o plano de ação em forma de 5W2H desenvolvido através dos conceitos *lean*.

A coleta de dados referentes a tempo e quantidade de retrabalhos, ocorreu a partir do monitoramento dos chamados em um *software* da empresa. Foram analisados os chamados com retrabalho no período de junho/2017 a setembro/2017, a partir de relatórios gerados pelo próprio

software, os quais listavam os chamados que geraram retrabalho e o tempo empregado no retrabalho.

4. Resultados

4.1 Processo Estudado

O ciclo do desenvolvimento de software inicia a partir da solicitação do cliente e termina na atualização do sistema do mesmo, contemplando a sua solicitação. Dessa forma, o chamado inicia no setor Suporte, é encaminhado para a Análise, em seguida para o Desenvolvimento e Teste, e por fim, retorna ao Suporte, como pode ser visualizado no fluxograma da Figura X

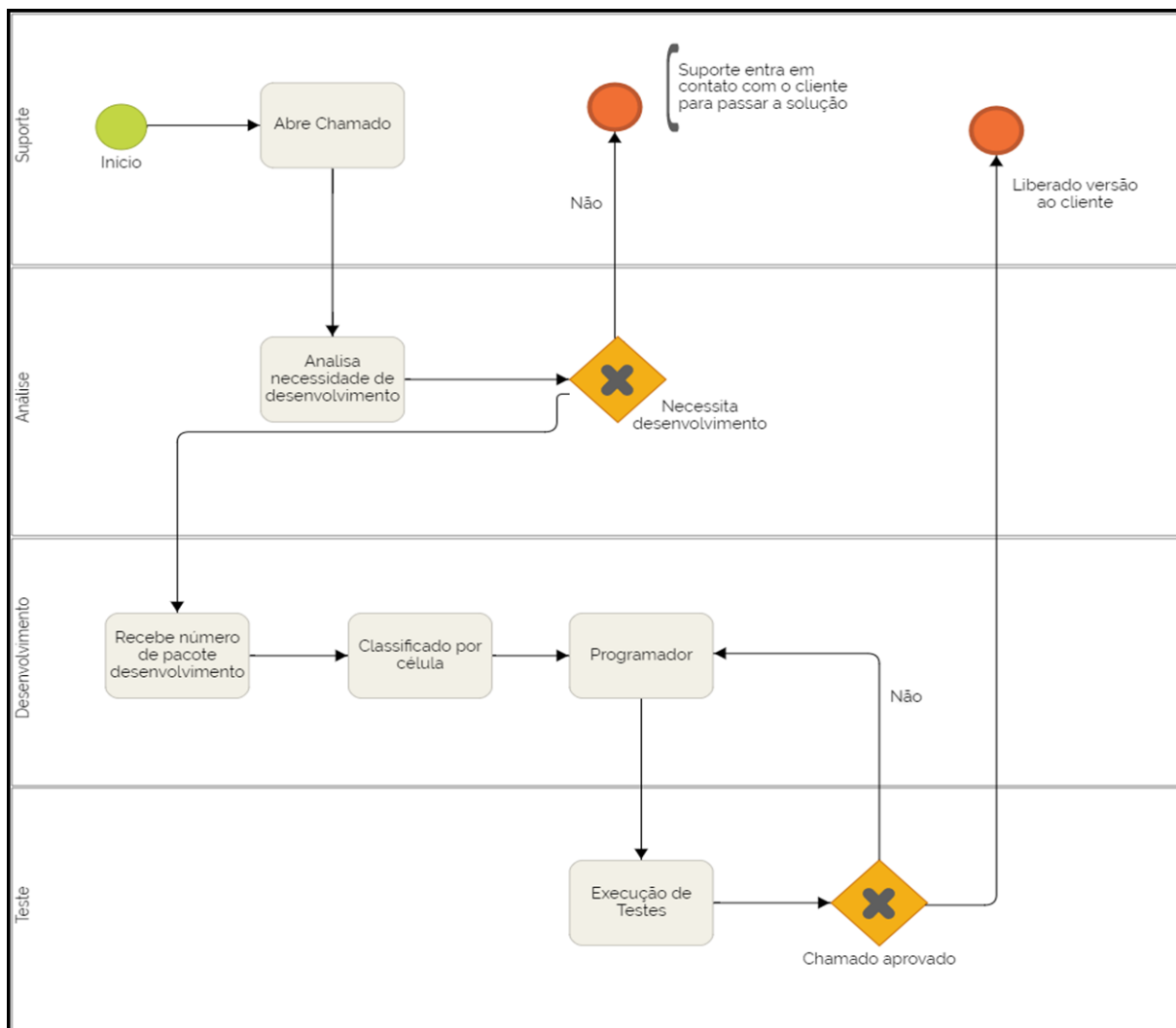


Figura 2 – Fluxograma do Ciclo do chamado

De acordo com a Figura 2, primeiramente o cliente entra em contato com o setor Suporte via telefone ou atendimento on-line e faz a sua solicitação de alteração no software, que pode ser uma melhoria, correção ou legislação. Neste momento, o analista de suporte abre um chamado no software de uso interno da empresa e documenta detalhadamente a solicitação do cliente com base nos seguintes itens: “Qual a ocorrência?”, “Onde ocorre a situação descrita?”, “Parecer técnico (processo para simular o descrito pelo cliente)”, “Cliente possui alguma particularidade?” e “Caminho do banco de dados para testes futuros”.

Após preenchido, o chamado é enviado para o usuário ‘Suporte’ no software de controle de chamados. O supervisor de Suporte ou Staff fazem a análise do preenchimento do chamado, acrescentam mais informações necessárias para um bom entendimento ou retornam o chamado para que o analista de suporte responsável por sua abertura faça uma melhor coleta de dados ou preenchimento.

Quando o chamado estiver com todas informações necessárias preenchidas, o mesmo é pontuado pelo supervisor de suporte ou staff, para então ser enviado ao usuário ‘Análise’. A pontuação do chamado é feita da seguinte forma: Impacto (alto, médio ou baixo); frequência de uso (atendimento, diário, semanal, quinzenal, mensal); tipo (parou de funcionar, erros, adequação) e stress do cliente (baixo, médio, alto, exorbitante). Essa pontuação serve para priorizar chamados, uma vez que os chamados com pontuação maior serão desenvolvidos com mais urgência do que os que possuem uma pontuação inferior.

O usuário ‘Análise’ trata-se do Gerente de Desenvolvimento, o qual fará a filtragem do chamado, descrevendo quais alterações precisam ser realizadas no software. A partir deste momento, o chamado entra em uma fila de espera que é determinada com base na pontuação e em prioridades estabelecidas pela empresa (exemplo, um chamado de erro na tela de faturamento é tratado com maior prioridade, assim como um chamado de implantação de legislação, o qual pode onerar multas para as empresas que não as cumprirem).

A cada 15 dias, são realizadas reuniões para planejamento de pacotes, onde os chamados que estão na fila são alocados em pacotes de atualização. Nessa reunião é definido qual célula trabalhará no desenvolvimento do chamado, além de determinar os programadores e testadores para cada chamado. É dimensionado inclusive quanto tempo será empregado para a programação e teste de cada chamado.

Após a reunião, o gerente de desenvolvimento encaminha cada chamado para a célula responsável, depois é direcionado para o programador responsável, conforme determinado na reunião de planejamento de pacotes.

O programador executa a solicitação descrita no chamado, criando configurações, telas, comandos, funções novas no software. Nesta etapa o programador precisa documentar os seguintes itens no software de controle de chamados: “A solicitação do cliente foi atendida?”, “O que foi realizado?”, “Foi alterado estrutura de banco de dados?”, “Foram alteradas ou acrescentadas configurações?”, “Foi criada nova tela?”, “Caminho de telas alteradas (caso novo)”, “Pode interferir em outros processos?”, “Houve alteração nas regras de negócio? O que mudou?” e “A alteração afeta outros sistemas?”.

Ao concluir a execução do chamado, o programador encaminha o chamado ao testador responsável, o qual verifica se o chamado está conforme as especificações e se o software não está retornando erros. Se forem identificadas falhas, o testador documenta e retorna o chamado para o programador realizar a correção. Caso estiver conforme, o testador realiza a documentação e conclui o chamado. Nesta etapa o testador precisa documentar os seguintes itens: “Problema inicial”, “Como foi resolvido? O que foi alterado?”, “Como está funcionando no momento após a alteração?” e “Configurações/Campos/Processos novos?”.

Após todos os chamados do pacote forem programados e aprovados pelos testadores, é gerado então um ‘upgrade’ que é o aplicativo atualizador do sistema, contendo todas as solicitações planejadas no pacote. Esse atualizador é executado pela automação de testes, o qual possui algumas sequências de rotinas de testes que são humanamente impossíveis de serem testadas rapidamente pelo testador. Se forem encontradas inconformidades, as mesmas são corrigidas,

caso contrário o executável é disponibilizado ao setor de Suporte.

Os chamados são dados como resolvidos pelos usuários que passaram, até retornar ao usuário de abertura. Esse usuário, que é o analista de suporte realizará testes do chamado em seu computador e se estiver conforme, entrará em contato com o cliente para realizar a atualização de seu sistema e explicar as alterações realizadas, bem como configurações e treinamentos se necessário. Caso o chamado estiver com inconformidades, será retornado ao desenvolvimento e ficará resolvido em um futuro pacote de atualização. O teste realizado pelo suporte, evita com que cheguem inconformidades até o cliente, o que causa descontentamento e falta de credibilidade.

4.2 Análise do problema

De acordo com o processo descrito no tópico 4.2, a observação do problema ocorreu no período de 01/06/2017 à 30/09/2017 e foi realizada com auxílio do *software* interno de controle de chamados da empresa, o qual listou através de relatórios os chamados classificados como TIPO “parou de funcionar”, além do tempo empregado no retrabalho.

Diante disso, viu-se a necessidade de criar uma planilha contendo a relação dos chamados que geraram retrabalho no período, para que após essa coleta de dados fosse possível realizar a devida análise do problema. Estas planilhas foram denominadas de folhas de verificação que são compostas pelos chamados que geraram retrabalho, o título do chamado, seu status, a sua pontuação e o tempo empregado para sua execução. Por exemplo, o chamado número 106301, refere-se a uma ocorrência na impressão de ordens de serviço, já está finalizado e foi implantado no cliente, possuía 184 pontos na lista de prioridades de chamados e demorou 26min40 para ser aberto, programado e testado.

A coluna SITUAÇÃO refere-se ao status do chamado no *software* de controle, uma vez que Pendente significa que o chamado apenas foi aberto pelo analista de suporte e ainda não foi desenvolvido; Resolvido significa que o chamado foi desenvolvido pelo programador, testado pelo testador e liberado em uma *sprint*, porém ainda não foi repassado para o cliente pelo analista de suporte; e o Implantado significa que o chamado já foi programado, testado, liberado e o analista de suporte já entrou em contato com o cliente para implantar a sua solicitação.

A coluna TEMPO foi feita com base no tempo em que o chamado foi aberto pelo analista de suporte, programado pelo programador, testado pelo testador e testado novamente pelo analista de suporte ao retornar para ele. Após a coleta de dados através das folhas de verificação, foi possível realizar uma análise do cenário atual da empresa e gerar indicadores referentes aos meses observados no estudo. Diante dessa premissa, o Figura 3 mostra um totalizador do número de retrabalhos em cada mês.

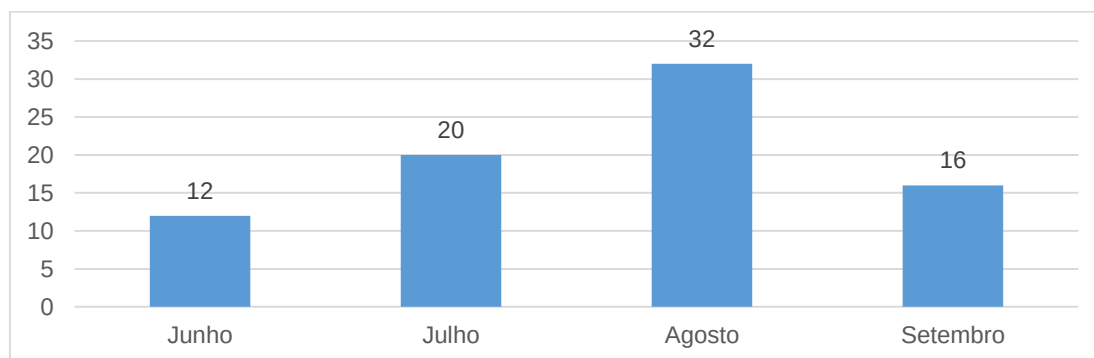


Figura 3 – Quantidade de chamados com retrabalho por mês

Dessa forma, conforme mostra a figura 3 chegou-se à conclusão de que entre junho/2017 e setembro/2017 80 chamados executados na fábrica foram em decorrência de retrabalhos, sendo 12 chamados em junho, 20 em julho, 32 em agosto e 16 em setembro. Diante desse fato e com base na coleta de dados da folha de verificação, foi possível criar outro indicador para expor a quantidade de tempo que os colaboradores empregaram em chamados que geraram retrabalho, conforme o Figura 4 a seguir.

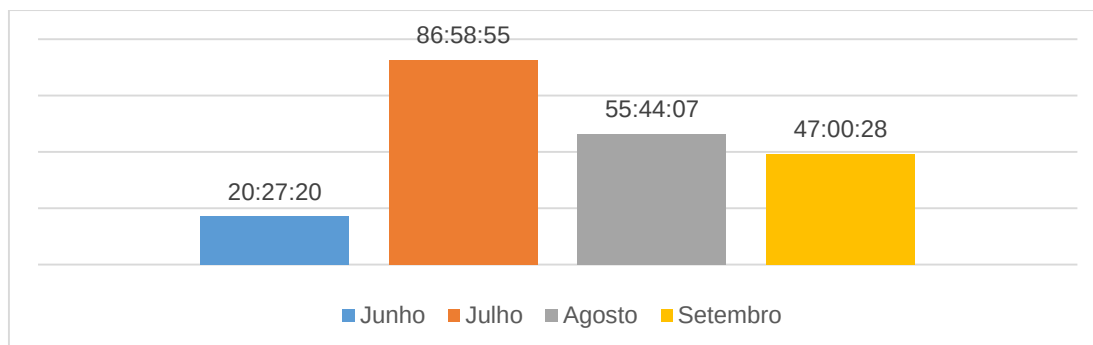


Figura 4 – Quantidade de tempo de retrabalho (hora : minuto : segundo)

Nota-se que o tempo empregado para correção de chamados de retrabalho em junho foi de 20:27:20, em julho 86:58:55, em agosto 55:44:07 e setembro 47:00:28, totalizando dessa forma em 210:10:50 no período analisado. O tempo varia de acordo com a complexidade do chamado e pelo tamanho da alteração que a mesma implica no sistema. A figura 5 mostra a quantidade de retrabalhos ocorridos em função do tempo no período analisado.

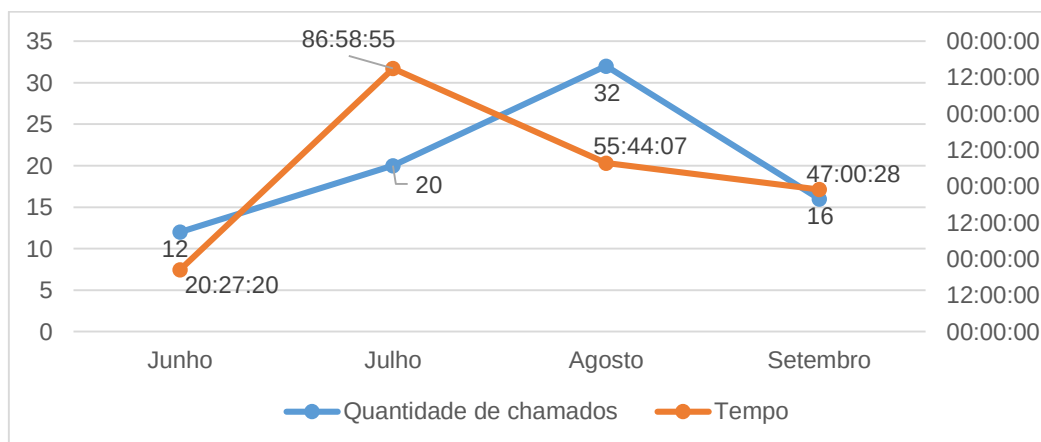


Figura 5 – Quantidade de tempo de retrabalho (hora : minuto : segundo)

Tendo base o Figura X, pode-se visualizar que o mês de junho gerou o menor números de retrabalhos no período, sendo 12 chamados. Em contrapartida agosto obteve o maior número, com 32 chamados de retrabalhos. Inclusive, nota-se que junho teve o menor tempo empregado para retrabalhos, sendo 20:27:20. Porém julho apresentou um elevado tempo de retrabalho, com 86:58:55 devido à complexidade dos chamados. Baseado na complexidade dos chamados, desenvolveu-se o seguinte indicador:

$$\text{Complexidade média global de chamados} = \frac{\text{tempo total}}{\text{número de chamados}}$$

Portanto, diante dessa equação foi possível desenvolver o Figura 5 para representar a complexidade média de chamados no período analisado.

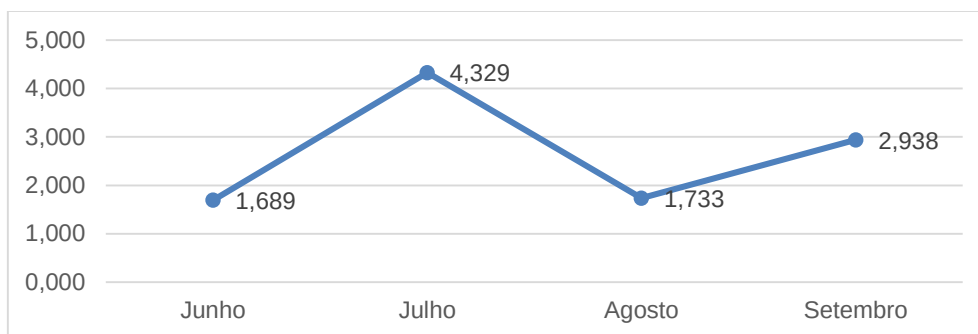


Figura 5 – Indicador de complexidade média global de chamados

Conforme a Figura 5, o mês de julho apresentou maior complexidade no desenvolvimento dos chamados, resultando em maior tempo de retrabalho que os demais meses. Já o mês de junho representou o menor indicador, coincidentemente houve a menor entrada de chamados de retrabalho e os mesmos não foram tão complexos quanto os demais meses, resultando assim em pouco tempo de execução. Nota-se que no período de junho/2017 a setembro/2017 a empresa teve 80 chamados de retrabalho, os quais totalizaram em 210:10:50 horas para a sua correção, envolvendo colaboradores do suporte, análise, programação e teste. Agosto foi o mês com mais entradas de retrabalho. Considerando que os colaboradores trabalhem 6 horas/dia, durante 4 meses e 22 dias/mês. Multiplicando-se os valores obtém-se que o tempo efetivo de trabalho foi de 528 horas no período analisado. Com base nos dados levantados, o total de tempo empregado em retrabalho resultou em 210:10:50, então para calcular o percentual de retrabalhos, foi desenvolvida a seguinte equação:

$$\% \text{ de retrabalho} = \frac{\text{Tempo de retrabalho}}{\text{Tempo efetivo de trabalho}} \times 100$$

Conclui-se então, a partir das equações que o percentual de retrabalho em função do tempo efetivamente trabalhado corresponde a 39,77%. Após identificar o tempo de retrabalho no desenvolvimento do *software*, foi levantado o total de remuneração do setor Suporte e Fábrica (desenvolvimento e teste) na empresa, resultando em R\$ 54.139,00. Com base nisso, chegou-se ao seguinte cálculo para identificar o custo direto do retrabalho:

$$\text{Custo imediato retrabalho} = \frac{\% \text{ de retrabalho}}{100} \times \text{total de remuneração}$$

Conclui-se então que o custo imediato de retrabalho gerado para a empresa no período foi de R\$ 21.531,08. Definiu-se este custo como ‘imediato’ devido ao valor considerar apenas o que a empresa remunera seus colaboradores para realizar o retrabalho. Não considera os custos indiretos como: consumo de energia elétrica, insatisfação dos clientes devido à demora na entrega, frustração dos colaboradores, entre outros. Após identificar o tempo e custo que o retrabalho onera para a empresa, faz-se necessário procurar identificar as possíveis causas que resultam neste. Desse modo, o tópico 4.6 abordará a análise das causas do retrabalho.

4.3 Análise do processo

A terceira fase da etapa P do ciclo PDCA, refere-se a análise do processo e relata as causas fundamentais do problema em questão. Essa análise se deu após a coleta de dados para

identificação do problema, com a realização de um *brainstorming* não estruturado com os colaboradores, os quais deram as suas opiniões e contribuições sobre as causas do retrabalho.

A pergunta realizada na reunião do *brainstorming* foi: “Por que certas funcionalidades param de funcionar no nosso sistema?”. As respostas obtidas foram: Empresa quer abranger vários seguimentos; Falta de análise de requisitos do software; Programador/testador fazem o que está sendo solicitado e não analisam o impacto nos demais processos que envolvem a alteração feita; falta de conhecimento de regras de negócios pelo programador; Alteração de uma funcionalidade pode interferir em outra; Má documentação dos chamados; Software muito amplo;

Observa-se nos itens abordados: falta análise de requisitos do software, programador/testador fazem o que está sendo solicitado e não analisam o impacto nos demais processos que envolvem a alteração feita, alteração de uma funcionalidade pode interferir em outra, má documentação dos chamados, estão diretamente relacionados com a fase análise de requisitos, teste e manutenção, do ciclo de vida de desenvolvimento de *software*.

A partir dos resultados obtidos na Figura 11, foi desenvolvido o diagrama de causa e efeito para identificar as principais causas para o problema retrabalho, conforme mostra a Figura 12.

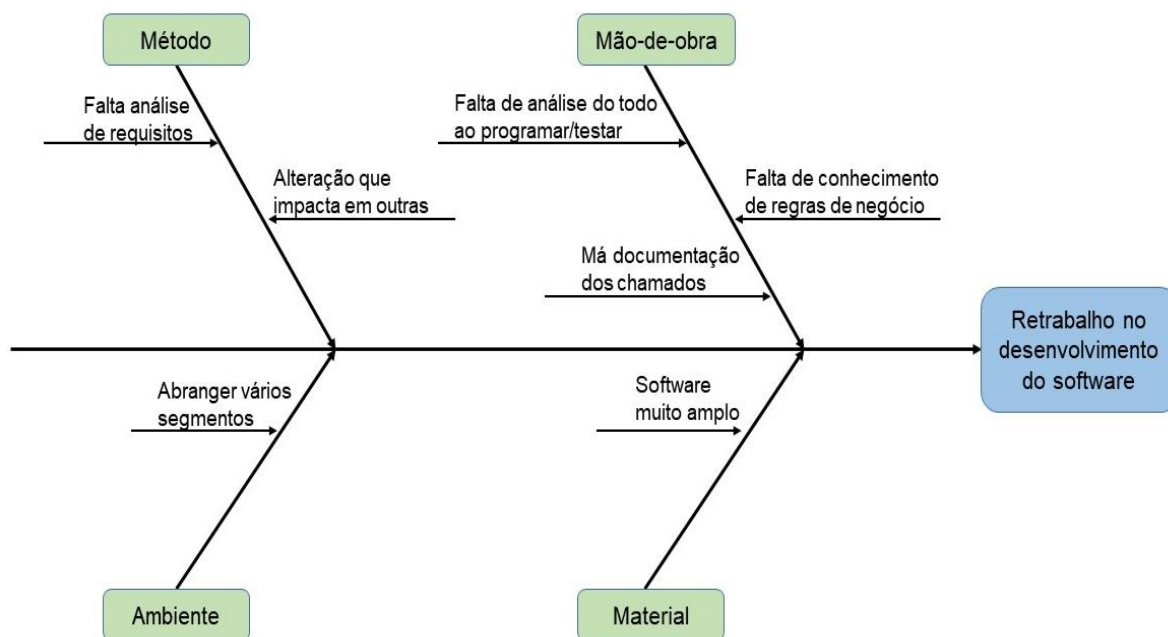


Figura 6 – Diagrama de causa e efeito

A figura 6 mostra o diagrama de causa e efeito, onde o problema é o Retrabalho no desenvolvimento do *software*, o qual está no traço horizontal. Aplicou-se a metodologia 4M (método, mão-de-obra, meio ambiente e material) com base nas afirmações levantadas com o *brainstorming*, as quais revelaram as possíveis causas para o problema em questão.

Com base no diagrama de causa e efeito, as causas mais prováveis para o retrabalho no desenvolvimento do *software* na empresa analisada, são: análise de requisitos, alterações que impactam em outras, falta de análise do todo ao programar/testar e má documentação dos chamados.

Após identificar as causas raiz do problema, é necessário desenvolver um plano de ação propondo melhorias para diminuir ou eliminar o problema constatado. Portanto, o tópico 4.7 abordará propostas de melhorias para o processo produtivo, visando a redução do retrabalho.

4.4 Plano de Ação

A quarta e última fase da etapa P do Ciclo PDCA refere-se na elaboração de um plano de ação para atacar a causa raiz. Portanto, após a identificação das causas mais prováveis para o problema, desenvolveu-se o plano de ação a partir da ferramenta 5W2H, conforme Quadro 8.

O que? (What?)	Quando? (When?)	Quem? (Who)	Onde? (Where)	Como? (How)	Quanto? (How much?)
Contratação de analista de requisitos	1ª semana de fevereiro/2018	Gerente da Fábrica	Fábrica	Entrevistas	R\$ 4.200,00
<i>Jidoka</i> – Automatização e <i>poka-yokes</i>	Segundo semestre de 2018	Analista de requisitos e testador	Fábrica	Rotina de trabalho	A avaliar
Disseminar a cultura <i>kaizen</i>	Toda sexta-feira do mês Janeiro/2018	Engenheira de produção	Empresa toda	Treinamentos e palestras	Sem custo

Quadro 1 – Plano de ação para redução do retrabalho

As propostas de melhorias para redução do retrabalho no desenvolvimento de *software* descritas no Quadro 01, foram baseadas nos conceitos da filosofia *lean* e na engenharia de *software*. A contratação de um analista de requisitos do *software*, o qual trabalhará no setor Fábrica (desenvolvimento e teste), resultará em uma melhor coleta dos dados e documentação dos chamados para não haver má interpretação. Além de especificar todas as operações, telas, configurações, regras de negócio, dentre outros, que devem ser levadas em consideração para a execução e teste de cada chamado.

Outro item do plano de ação, refere-se a automatizar (*jidoka*) o restante do *software*, pois hoje apenas uma parte do teste é automatizado. Essa automatização deve ser feita por um testador em conjunto com o analista de requisitos, pois todos os requisitos devem ser automatizados, para que não se corra o risco de passar despercebido e gerar um futuro retrabalho. Inclusive incluir *poka-yokes* no código fonte do sistema e na automação de testes, que são travas sobre bugs ou situações que possam gerar o retrabalho.

E o último item do plano de ação, refere-se à disseminação da cultura *kaizen* na empresa, através de treinamentos que irão frisar a importância da melhoria contínua no ambiente de trabalho, dinâmicas que possibilitarão ouvir a opinião dos colaboradores e por fim, ensinar como aplicar o Ciclo PDCA no dia-a-dia de trabalho, para resolução de problemas. Essa etapa tem por objetivo, despertar o interesse nos colaboradores pela busca da melhoria contínua, fazendo com que as pequenas mudanças se tornem importantes para o processo como um todo. Observa-se que os itens propostos, fundamentaram-se principalmente nos princípios da filosofia *lean* e *lean* de *software*: eliminar o desperdício, integrar qualidade e criar conhecimento.

5 CONSIDERAÇÕES FINAIS

Este trabalho fundamentou-se na análise do retrabalho no desenvolvimento de *software*, bem como a criação de indicadores para a quantificação de custos relacionados ao retrabalho e proposta para a sua redução.

Primeiramente, procurou-se mapear o processo produtivo de engenharia de *software* para levantamento de custos relacionados ao retrabalho. Identificou-se que no período de junho/2017 a setembro/2017 a empresa obteve 80 chamados de retrabalho, os quais totalizaram em 210:10:50 horas, resultando em um percentual de 39,77% de retrabalho. Após o levantamento de custos, foi possível concluir que o retrabalho gerou um custo de R\$ 21.531,08 para a empresa. Diante dos números obtidos, percebe-se que o retrabalho é um desperdício no processo produtivo e deve ser estudado de forma a ser reduzido. Dessa forma o colaborador poderá desempenhar as suas funções em atividades que agregam valor ao produto, diminuindo o custo e maximizando os resultados.

Finalmente, procurou-se adequar o modelo de produção *lean* para a engenharia de *software* da empresa, bem como propor melhorias para a redução do retrabalho, as quais estão especificadas no item 4.4 e fundamentaram-se principalmente nos princípios *lean*: eliminar o desperdício, integrar qualidade e criar conhecimento.

Conclui-se então que a Engenharia de Produção diante de suas atribuições técnicas, proporciona o estudo e análise do processo produtivo da Engenharia de *Software* e possibilita contribuições significativas através de propostas de melhorias baseadas nos conceitos vistos ao longo da formação acadêmica.

Propõe-se para trabalhos futuros a implementação e controle do plano de ação proposto, além de implantar o mapeamento de fluxo de valor proposto pelo *lean* de *software*, para identificação de outros desperdícios que ocorrem no processo, bem como dar suporte para a diminuição ou eliminação dos mesmos.

Referências

- BARTIÉ, A.** *Garantia da qualidade de Software: adquirindo maturidade organizacional*. Rio de Janeiro: Elsevier, 2002.
- BOEHM, B.; BASILI, V.** *Software Defect Reduction Top 10 List*. *IEEE Computer*. v. 34, n.1, p. 135-137, 2001.
- CORRÊA, H. L.; CORRÊA, C. A.** *Administração da produção e operações: manufatura e serviços: uma abordagem estratégica*. 2. ed. São Paulo: Atlas, 2011.
- JUNIOR, C. C.** *Sistemas integrados de gestão - ERP: uma abordagem gerencial*. 3. ed. Curitiba: Ibpex, 2008.
- KALINOWSKI, M.; SPINOLA, R. O.** *Introdução à Inspeção de Software*. *Engenharia de Software Magazine*. v. 1, n.1, p. 68-74, 2008.
- KOSCIANSKI, A; SOARES, M. S.** *Qualidade de Software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. 2. ed. São Paulo: Novatec Editora, 2007.
- LIKER, J. K.; MEIER, D.** *O Modelo Toyota: Manual de aplicação*. Porto Alegre: Bookman, 2007.
- LUSTOSA, L. et al.** *Planejamento e controle da produção*. Rio de Janeiro: Elsevier, 2008
- MERHI, D.** *40+10 ferramentas e técnicas de gerenciamento*. Rio de Janeiro: Brasport, 2013.



VIII CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO

Ponta Grossa, PR, Brasil, 05 a 07 de dezembro de 2018

MOREIRA, D. A. *Administração da Produção e Operações*. 2. ed. São Paulo: Cengage Learning, 2011.

POPPENDIECK, M.; POPPENDIECK, T. *Implementando o desenvolvimento Lean de software: do conceito ao dinheiro*. Porto Alegre: Bookman, 2011.

RODRIGUES, M. V. *Ações para a qualidade GEIQ: Gestão integrada para a qualidade: padrão Seis Sigma, classe mundial*. 2. ed. Rio de Janeiro: Qualitymark, 2006.

RODRIGUES, M. V. *Entendendo, aprendendo e desenvolvendo sistema de produção Lean Manufacturing*. 2. ed. Rio de Janeiro: Elsevier, 2016.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da produção*. 2. ed. São Paulo: Atlas, 2002.

SOMMERVILLE, I. *Engenharia de Software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

TONSIG, S. L. *Engenharia de Software: Análise e Projeto de Sistemas*. 2. ed. Rio de Janeiro: Editora Ciência Moderna LTDA., 2008.

TUBINO, D. F. *Planejamento e controle da produção: teoria e prática*. São Paulo: Altas, 2007.

VERGUEIRO, W. *Qualidade em serviços de informação*. São Paulo: Arte e ciência, 2002

VIEIRA, S. *Estatística para a qualidade*. 2. ed. Rio de Janeiro: Elsevier, 2012.

WERKEMA, C. *Design for Lean Six Sigma: ferramentas básicas usadas nas etapas D e M do DMADV*. Rio de Janeiro: Elsevier, 2012.

WERKEMA, C. *Criando a Cultura Lean Seis Sigma*. Rio de Janeiro: Elsevier, 2012.

WOMACK, J. P.; JONES, D. T.; ROSS, D. *A Máquina que Mudou o Mundo*. 2ª. ed. Rio de Janeiro: Campus, 2004.