20° Automação & TI
ISSN: 2237-0234

UTILIZAÇÃO DE MÉTODOLOGIAS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE NA MELHORIA CONTÍNUA DE PROCESSOS INDUSTRIAIS*

Pedro Sampaio Cotta[1]
Gabriel Andrade Medeiros[2]

Resumo

O mercado de Siderurgia no Brasil vive um momento de pressão causado pelas quedas dos preços internacionais e agravado por fatores internos como a redução no consumo, aumento do custo de energia e tributação elevada. Tendo em vista que, ao menos no curto prazo, não há perspectiva de melhoria, as indústrias do setor se mobilizam para adequar a esta nova realidade. A sobrevivência em um cenário de alta competitividade é garantida reduzindo custos de produção e manutenção e buscando aumentar a produtividade sem, entretanto, expandir a capacidade produtiva. Para atingir estes objetivos as indústrias focam em pequenas melhorias operacionais, com investimentos em pequenas modernizações e otimizações de processos existentes, em detrimento de grandes investimentos em expansões. Neste contexto observa-se um aumento na relevância do software de automação, que surge como alternativa para realização de melhorias, pois possuem custo relativamente baixo em comparação com reformas a nível mecânico e podem proporcionar ganhos importantes de eficiência e qualidade. Este artigo apresenta um estudo da utilização de metodologias ágeis de desenvolvimento de software para realização de melhoria contínua em processos industriais. Como exemplo de aplicação é apresentado o trabalho de otimização do software de automação de um laminador de alumínio com foco no aumento de sua taxa de utilização. A meta foi buscar um ganho de produtividade garantindo os mesmos padrões de segurança originais dos equipamentos e com impacto mínimo nas atividades produtivas.

Palavras-chave: Automação; Melhoria contínua; Metodologias ágeis; Primetals Technologies Brazil.

USE OF AGILE SOFTWARE DEVELOPMENT FOR CONTINUOUS IMPROVEMENT OF INDUSTRIAL PROCESSES

Abstract

The Iron and Steelmaking market in Brazil is experiencing a moment of pressure caused by the drop of international prices and aggravated by internal factors like the reduction in consumption, the increased cost of energy and high taxation. Given that, at least in short term, there's no prospect of improvement, the sectors industries are mobilizing to fit this new reality. Survival in a highly competitive scenario is guaranteed reducing production and maintenance costs and seeking to increase productivity without, however expanding production capacity. To achieve these goals the industries focus on small operational improvements, with investments in small optimizations and modernizations of existing processes, rather than large investments in expansions. In this context an increase in the relevance of automation software is observed, which becomes an alternative for making improvements because they have relatively low cost compared to mechanical revamps and can provide significant gains in efficiency and quality. This article presents a study of the use of Agile software development for achieving continuous improvement in industrial processes. As an example of application is presented the work of automation software optimization of an aluminum rolling mill with a focus on increasing its utilization rate. The goal was to seek a productivity gain guaranteeing the same original safety standards of equipment and with minimal impact on productive activities.

Keywords: Automation; Optimization; Process Control; Primetals Technologies Brazil.

[1] *Engenheiro de Projetos, PEP AUT, Primetals Technologies, Belo Horizonte, MG, Brasil.*
[2] *Coordenador de Engenharia, PEP AUT, Primetals Technologies, Belo Horizonte, MG, Brasil.*

1 INTRODUÇÃO

Existem inúmeras evidências do aumento da complexidade e crescimento da importância do software nos sistemas de Automação Industrial [1]. Em um estudo, a Federação de Engenharia Alemã constatou que o percentual médio dos custos de desenvolvimento de software presentes no valor total de um maquinário, cresceu de 20% para 40% na ultima década [2]. Em adição, a crescente concorrência e demanda do mercado por produtos customizados, tem levado as indústrias a uma corrida de inovação para tornar suas linhas de produção flexíveis e capazes de se adaptar a novas necessidades do mercado [3]. Devido ao elevado custo associado à remodelação de maquinário, prefere-se então investir em alterações de software e como consequência, tem se observado um crescente interesse de em se aplicar práticas de engenharia de software, principalmente práticas ágeis, que possuem comprovado sucesso de sua utilização em uma variedade de cenários [4].

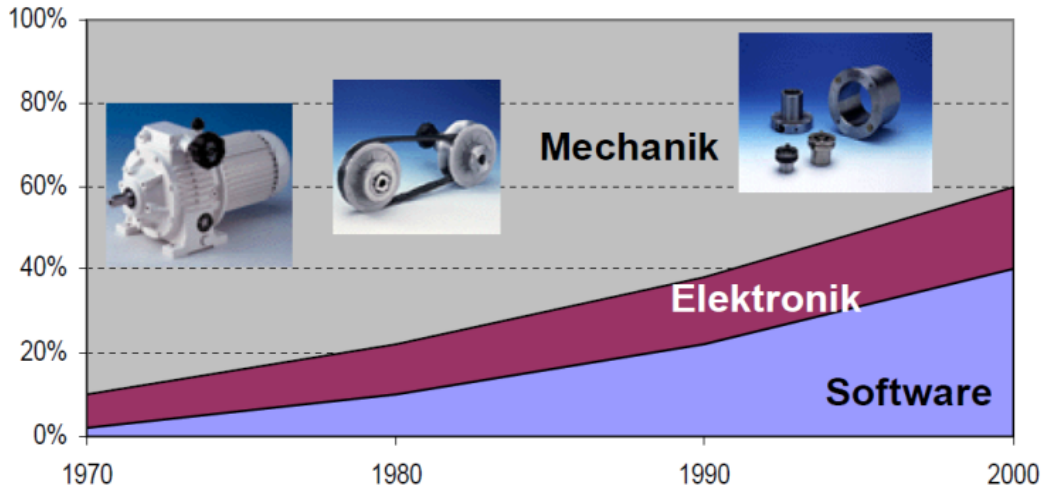

Figura 1. Divisão dos custos em projetos industriais [2].

Outro ponto que contribui para o aumento da importância do software é a ampliação da adoção da *Internet of things (IoT)*. No meio industrial isso significa que, cada vez mais equipamentos e sensores possuem software embarcado com intuito de trocar informações ou realizar lógicas de controle [5].

Apesar do crescente interesse, a escassez de estudos consolidados sobre a utilização de práticas ágeis, especificamente em projetos de controle de equipamentos, torna necessário compreender como estes métodos podem ser aplicados e quais benefícios podem ser almejados. Atualmente a maioria das pesquisas realizadas sobre o tema se baseia em um contexto típico, no qual os *frameworks* e ferramentas estão adaptados de forma a facilitar a aplicação das técnicas [6].

Neste estudo mensuraram-se vantagens e desvantagens da aplicação de Metodologias Ágeis especificamente no contexto da prestação de serviços de Automação Industrial. A Primetals Technologies Brazil foi contratada pela Novelis do Brasil para participar da equipe responsável pela melhoria contínua nos processos de automação do seu laminador CM3 localizado em Pindamonhangaba, São Paulo. O projeto, denominado Kaizen, contemplou a identificação de melhorias que impactariam na redução do tempo ocioso para troca de bobinas no laminador, resultando consequentemente em um aumento de eficiência.

2 MATERIAIS E MÉTODOS

2.1. Métodos Ágeis

O conceito de métodos ágeis foi inicialmente introduzido no meio de desenvolvimento de software através do Manifesto Ágil [7]. Essencialmente o que se propõe é priorizar a entrega de valor através de entregas antecipadas e contínuas de software, trabalhando de forma integrada e colaborativa e abraçando mudanças mesmo que tardias. A tabela abaixo sintetiza as principais diferenças de métodos ágeis e práticas tradicionais de desenvolvimento de software.

Tabela 1- Comparando métodos de desenvolvimento de software tradicional e ágil [8].

Propriedades	Tradicional	Ágil
Atitude	Preditiva	Adaptativa
Tamanho de projeto	Grande	Pequeno
Tamanho e mentalidade do time	Grande / Disciplinado	Pequeno / Inovador
Modelo de gerenciamento	Autocrático	Descentralizado
Atitude em relação a mudanças	Resistente	Abraça
Documentação	Compreensiva	Leve e abstrata
Planejamento de longo prazo	Compreensivo	Limitado
Ciclo de vida	Preso e limitado	Iterativo e ilimitado
Cultura organizacional	Comando e controle	Liderança e colaboração
Retorno de investimento	Ao final do projeto	Em estágios iniciais

2.1.1 Scrum

Scrum é o framework ágil mais popular e mais aplicado no meio de desenvolvimento de software [9] e sua utilização não fica restrita a área de software. Scrum é adequado para qualquer escopo complexo e ou inovador de trabalho [10].

Hirotaka Takeuchi e Ikujiro Nonaka, ao realizar uma analogia com o rúgbi, apresentaram Scrum como método de desenvolvimento de produto, em um artigo publicado em Harvard, no ano de 1986. A intenção era demonstrar que, por meio do trabalho em equipe e da colaboração, melhores resultados podem ser obtidos no ambiente competitivo em que vivemos [11].

Scrum geralmente é referenciado como um framework fácil de ser entendido, porém difícil de ser aplicado. Quando se implementa Scrum, inicialmente uma pessoa no papel de *Product Owner* cria e prioriza uma lista denominada *Product Backlog*. Durante o planejamento de uma *Sprint* a equipe do projeto seleciona os itens de maior prioridade do *Product Backlog* e idealiza como implementá-los. A *Sprint* mencionada anteriormente define um intervalo de tempo (entre uma e quatro semanas tradicionalmente) para que a equipe realize os itens selecionados. Durante esse tempo o time se reúne diariamente para avaliar o progresso e relatar eventuais impedimentos em uma reunião curta. Uma pessoa no papel de *Scrum Master* atua ao longo da *Sprint* com objetivo de manter a equipe focada e livre de empecilhos para cumprir com o planejado. Ao final de uma *Sprint* deve haver uma parte do

projeto passível de ser entregue, ou seja, implementado e testado, a ser disponibilizado ao cliente. A *Sprint* é encerrada com uma reunião de revisão e retrospectiva. Idealmente este ciclo se repete até que o *Backlog* seja totalmente realizado ou até que o orçamento seja totalmente consumido. Desta forma se garante maior entrega de valor possível ao cliente independentemente de orçamento ou prazo.

Figura 2 - Processo Scrum simplificado [10].

Por ser um framework iterativo e incremental, Scrum permite o rastreamento de perto do projeto e provê resultados imediatos. Com Scrum a mudança de requisitos é permitida e realizada de forma mais ágil. Por fim um dos maiores benefícios relacionados ao Scrum é a melhoria de comunicação e relacionamento entre os membros da equipe e o cliente.

2.1.2 Kanban

Kanban é uma técnica *Lean* utilizada na administração de produção, derivada do *Just In Time*, que permite agilizar a entrega e produção de produtos. O foco está na entrega contínua. Uma das características mais reconhecidas do Kanban é o conceito *WIP* (*work in progress*) que limita o número de tarefas em execução simultaneamente [12].

Assim como no Scrum, os membros da equipe aplicando Kanban trabalham individualmente na execução de tarefas e podem assumir uma nova tarefa assim que necessário. Em Kanban não existem papeis como no Scrum e não há comprometimento de se realizar uma reunião diária. De forma geral seu processo é mais flexível. Scrum e Kanban podem ser combinados e aplicados sob a forma de *Srcrumban* [12]. Neste modelo é combinada a flexibilidade do Kanban com os atributos previstos no Scrum. Uma equipe que utiliza Scrum pode, por exemplo, utilizar quadros Kanban para praticar gestão a vista para manter o time atualizado sobre em quê cada um está trabalhando. Para evitar que membros da equipe fiquem sobrecarregados limita-se o número de tarefas que cada um pode realizar (*WIP*) com o intuito de balancear a carga de trabalho e favorecer a entrega contínua.

2.1.3 Extreme Programming

Enquanto Scrum é um *framework* cujo foco está no gerenciamento de projetos [13], Extreme programming (XP) enfatiza o desenvolvimento de Software. Os principais princípios defendidos em XP são [14]:

- Codificação como atividade principal;
- Cultura de *Just In Time* com ciclos de *Release* frequentes;
- Ciclo de vida e comportamento definidos em Casos de Teste;
  - Design simples para solução de problemas e refatoração quando necessário.

No XP quando se propõe utilizar ciclos de Release frequentes (dias, horas ou minutos ao invés de semanas ou meses) e combinando com *Just In Time*, o resultado é que só se produz código quando necessário, e as funcionalidades são disponibilizadas ao cliente assim que realizadas. Isto reduz drasticamente o tempo que uma funcionalidade demora até chegar ao mercado.

Em XP o desenvolvimento é orientado a buscar as soluções mais simples possíveis. A prática de tentar prever necessidades futuras e se antecipar para atendê-las é desencorajada. Em compensação, se busca revisar o código a todo o momento e refatorar quando necessário. A arquitetura do sistema deve ser trabalhada e redefinida sempre que houver necessidade.

Uma prática comum em XP, que colabora para a produção de código de qualidade, é a programação em pares. A ideia é que, enquanto uma pessoa desenvolve, outra pessoa revisa o código gerado em tempo real focando em qualidade. Cockburn e Williams verificaram que com um esforço extra de 15% no desenvolvimento houve melhoras no design, redução de defeitos, desenvolvimento das habilidades do time e maior eficiência na comunicação. Além disso, praticar XP foi considerado mais agradável em relação a codificar individualmente [15].

2.1.4 Test Driven Development

Test driven development (TDD) é uma prática ágil, que emergiu do XP e que induz o desenvolvedor a produzir código somente para satisfazer as condições de um caso de teste. Para cada funcionalidade a ser criada, inicia-se concebendo um teste que valida o objetivo do código. Inicialmente como não há código implementado, o teste irá falhar. Em seguida o desenvolvedor produz o código somente para satisfazer as condições do teste. Após a funcionalidade estar realizada, tanto o código da aplicação quanto o do teste podem ser refatorados se necessário [16].

A principal diferença que TDD introduz em relação ao modelo em cascata é a especificação dos casos de teste antes da funcionalidade ser implementada. Entretanto TDD é mais que isso. Utilizando TDD, além de o teste ser utilizado para ditar que código será produzido, o caso de teste serve também como especificação. Alguns autores argumentam que o teste faz parte da documentação e em muitos ambientes ágeis o testador também participa das etapas de análise, definição e priorização dos requisitos. O benefício de utilizar TDD é que erros são identificados precocemente e os sistemas gerados são mais isentos de falhas, principalmente devido à cobertura ampla dos testes de unidade. Com TDD fica mais fácil ser adaptável a mudanças e adições de funcionalidades.

Um exemplo de caso de sucesso é a comparação realizada na IBM por Williams e Maximilien [17]. Ao adotar TDD em um projeto em andamento observou-se uma redução de 40% na detecção de erros pela equipe de testes externa com uma produtividade semelhante a anterior. A justificativa para este fenômeno foi que

apesar do tempo gasto escrevendo testes de unidade, a equipe observou uma diminuição na necessidade de depurar o código para adequar uma funcionalidade.

2.1.5 Utilização de Métodos Ágeis em Sistemas de Automação Industrial

Em Automação Industrial praticamente todos os grandes fabricantes de CLP's, com Siemens, Schneider, GE e ABB, adotam o padrão IEC 61131 em linguagens de programação para desenvolvimento de software [18]. Este padrão não define a realização da orientação a objeto de forma completa logo está mais alinhado ao preconizado no modelo cascata de desenvolvimento. Contudo atualmente algumas iniciativas têm voltado suas atenções para práticas ágeis. Um exemplo é a ultima revisão do modelo GxP, utilizado pela indústria farmacêutica. O guia revisado em 2008 cita programação ágil de forma explicita:

> "The approach described in this document is designed to be compatible with a wide range of other models, methods, and schemes, including: Software development methods such as RAD, Agile, RUP or Extreme programming." [19].

No artigo foi realizado um estudo das seguintes práticas ágeis: *Scrum*, *Extreme Programming*, *Iterative Conference Room Pilot* e *Model Driven Architecture*. A aplicabilidade das práticas em relação ao modelo GxP foi analisada. Em conclusão, foram citados como pontos positivos: a utilização de passos iterativos e incrementais para se chegar a uma solução; o teste das funcionalidades assim que implementadas; o envolvimento antecipado e constante da comunidade e a flexibilidade ao lidar com mudanças de requisitos e especificações.

Outra referência é a utilização de métodos ágeis para reconstrução de um supervisório [20]. No artigo é citado que projetos de supervisórios falham principalmente por: a especificação ser pouco detalhada; tempos de implementação serem longos; o limitado envolvimento do operador do sistema e as expectativas em relação ao projeto não serem gerenciadas. Comportando-se como aconselhado em métodos ágeis como: promover a colaboração e envolvimento dos *stakeholders* e não tentar tomar todas as decisões antecipadamente foram pontos importantes para o sucesso do projeto.

Quando se refere a testes para sistemas que executam em CLP's não é comum observarmos a programação de testes de unidade e execução automatizada de casos de teste. Testes de unidade em sistemas de automação são mais complicados de serem realizados, pois deve ser considerada a interação com hardware [21]. O que se vê geralmente é a baixa automação de testes o que acarreta em custos elevados durante o teste de integração. O artigo propõe a técnica denominada *Test Driven Automation* que pressupõe desenvolver o teste antes de implementar a funcionalidade. Em continuidade ao recomendado um modelo sistemático para realização de teste ágil em consonância com as normas IEC 61131-3 e IEC 61499 foi proposto [22]. O modelo suportaria inclusive a execução automática de testes. Em um trabalho futuro os autores esperam utilizar testes baseados em modelos UML para geração automática de casos de testes.

2.2 Identificação de pontos de melhoria

O projeto Kaizen foi idealizado para o Tandem Cold Mill 3, o mais recente laminador a frio da Novelis, comissionado em 2012 pela Primetals Technologies na sua planta em Pindamonhangaba – SP. Para execução do projeto foi formada, pela Novelis, uma equipe composta de profissionais de capacidades variadas. O grupo contou

com a experiência em desenvolvimentos de projetos de elétrica e automação da Primetals Technologies através de um engenheiro de automação na planta além de suporte no escritório. A participação de Stakeholders de variadas posições foi essencial para garantir a execução de forma natural e concomitante com a operação normal de produção da planta. O envolvimento da Primetals se deu principalmente para que as modificações fossem realizadas de forma segura, com o mínimo de intervenção na operação normal de produção.

Projetos em que se utilizam metodologias ágeis têm como objetivo principal garantir a maior entrega de valor possível ao cliente dado um orçamento e prazo. Para o projeto Kaizen, entregar valor significou reduzir o tempo entre bobinas ou TBC (*time between coils*). TBC é o tempo de ociosidade do laminador, medido a partir do momento em que uma bobina acaba de ser laminada até o ponto em que uma nova bobina é alimentada e um novo regime de laminação é alcançado. Neste período, especificamente para o laminador em questão, as seguintes atividades são executadas:

- Desaceleração ao final da laminação;
- Alivio da tensão, abertura das cadeiras e corte do material pela tesoura de saída;
- Enrolamento do material acabado no mandril de saída e do material não laminado no mandril de entrada;
- Remoção da bobina acabada e alimentação de um novo carretel no mandril de saída;
- Remoção do material não laminado na entrada do laminador e admissão de uma nova bobina;
- Preparação da ponta do material a ser laminado;
- Alimentação do material pelas cadeiras de laminação engate da ponta no mandril de saída;
- Aceleração e redução de espessura para atingir as especificações de regime de laminação.

Dentre as atividades listadas somente a preparação da ponta é uma atividade manual. O restante é governado por controles automáticos e consequentemente passível de melhoria de software.

O projeto foi iniciado com uma reunião para apresentação e levantamento de requisitos. As principais melhorias a serem realizadas foram listadas pelos participantes. A partir desta lista, contando com a avaliação dos membros da equipe, foi produzida uma matriz esforço x benefício. Desta forma foi possível selecionar, dentre as atividades listadas, quais apresentam maior potencial e consequentemente deveriam ser realizadas de forma prioritária. Responsáveis foram definidos para cada item da lista de ações e o acompanhamento do avanço realizado semanalmente.

ISSN: 2237-0234

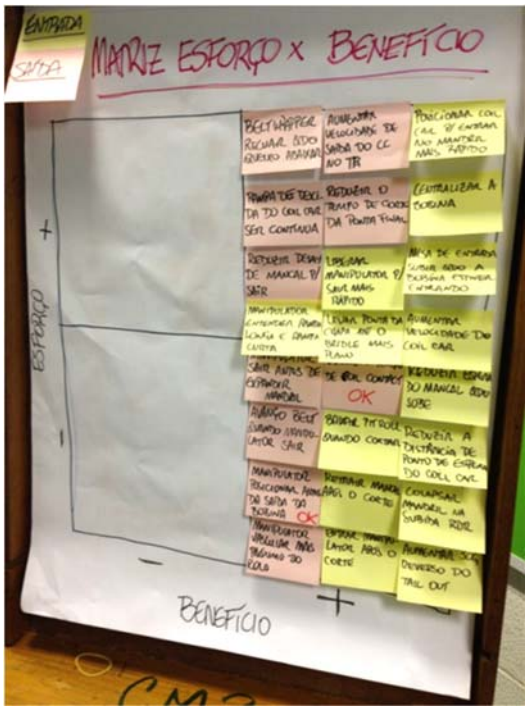

Figura 3. Atividades do projeto em uma matriz esforço x benefício.

Sobre o levantamento de requisitos, é importante salientar que a lista de ações não foi fechada após o Brainstorming inicial. Pontos importantes foram levantados durante a execução do projeto, principalmente através de conversas com operadores mais experientes e que conhecem profundamente o funcionamento do laminador, pois vivenciam sua operação diariamente.
Analisando-se a lista de melhorias propostas, verificou-se que a maior parte das ações levantadas compreendiam alterações de software e que estas ações têm como característica um bom custo/benefício. Isto se deve ao fato de que as modificações de software geralmente requerem esforço e custos relativamente inferiores em relações a alterações mecânicas, pois na maioria dos casos não envolvem compra de equipamentos e não requerem grandes paradas da produção.

2.3 Arquitetura dos sistemas de controle

O controle de processos do laminador é realizado através de controladores Siemens. Um rack TDC contendo sete CPU´s realiza o controle de funções tecnológicas e de coordenação enquanto que três racks S7-400 controlam os equipamentos individualmente e realizam outras funções auxiliares ao processo. Além disso, há um rack S7-400F responsável exclusivamente pelas funções de segurança do laminador.
A comunicação entre os controladores e os equipamentos, para a maioria dos sinais e acionamentos, é facilitada com a utilização de remotas de campo ET200S e comunicação Profibus. Entre os controladores e o sistema de supervisão a troca de dados é realizada por protocolos TCP/IP e UDP via Industrial Ethernet. A programação do software de automação do Nível 1 é realizada com o Simatic PCS7 para todos os controladores e as funções de Nível 2 através do Simatic WinCC.
A arquitetura dos sistemas de chão de fábrica do laminador é benéfica para manutenção, pois, entre outros benefícios, ajuda na redução do tempo necessário para paradas. O hardware permite que os módulos de sinais sejam alterados com o

sistema em operação e seu design modular permite uma capacidade de periférica virtualmente ilimitada [23]. Em relação ao software, a solução fornecida também é benéfica para a realização das modificações, pois o conjunto PCS7/WinCC permite uma análise simples e transparente do software implementado além de proporcionar a realização de alterações de forma rápida e eficiente sendo que na maioria dos casos não houve necessidade de interromper as atividades produtivas para efetivação de uma alteração nos sistemas.

2.4 Desenvolvimento e testes

Uma característica importante inerente ao desenvolvimento de sistemas de automação é que, apesar destes serem desenvolvidos em avanço, é durante a fase de comissionamento que várias funcionalidades são ajustadas para adequação às condições do processo [24]. Este comportamento é observado porque certos controles só podem ser corretamente configurados quando o maquinário associado estiver instalado. Em alguns casos há a possibilidade de realizar simulações, porém é custoso reproduzir o comportamento real do processo. Como consequência é necessário lidar com alterações de software diretamente no ambiente de produção.
As adaptações nas lógicas relacionadas ao TBC eram realizadas paralelamente as atividades de produção, em horário comercial e com acompanhamento da equipe de manutenção elétrica. Os períodos de parada preventiva também eram aproveitados para testes e desenvolvimentos que não poderiam ser realizados sem perturbar a operação. Dependendo da funcionalidade modificada, testes também ocorriam com o laminador em funcionamento. Otimizações em controles PID, por exemplo, eram realizadas de forma gradativa com o intuito de não causar variações bruscas em variáveis de processo.
Considerando que as modificações no software eram realizadas e testadas no ambiente de produção, foi importante garantir a normalidade nas atividades de produção, principalmente nos turnos da noite/madrugada e finais de semana, quando a presença de engenheiros de manutenção era reduzida. Desta forma foi acordado que ao final do dia as alterações que não foram suficientemente testadas eram revertidas para a lógica original. Uma alteração era homologada somente após a realização de testes em cenários variados e de forma prolongada. Através da realização do controle de alterações, era possível retornar uma funcionalidade ao comportamento original caso algo não ocorresse da forma desejada.
Foram realizadas mais de 30 alterações de software com objetivo de melhorar o desempenho da troca de bobinas. Listando de forma resumida, destacam-se as seguintes iniciativas:

- Identificação e correção/melhoria de lógicas de equipamentos que apresentavam falhas com maior frequência;
- Reorganização no sistema de mensagens de alarmes. Eliminando mensagens irrelevantes e criando mensagens de alerta para evitar necessidade de paradas;
- Exibição de novas informações no sistema supervisório para diagnostico de anormalidades de processo;
- Otimizações de controles PID para acelerar o movimento de equipamentos e estabilização de variáveis do processo;
- Alteração de passos em sequencia automática, principalmente para realização de movimentos em paralelo;
- Antecipar posicionamento de equipamentos na posição de laminação;

- Criação do Kaizen Online que permitiu o acompanhamento em tempo real do TBC além da visualização da média mensal por turma.

2.5 Acompanhamento de avanço

Neste projeto foram empregadas práticas e técnicas ágeis para gerenciamento e acompanhamento das atividades. A lista de atividades, definida em conjunto com a Novelis, foi utilizada como *Product Backlog* para definição de *Sprints* semanais. Em média eram necessárias duas horas semanais para planejamento é acompanhamento da sprint.
Conforme referenciado no *Scrum* os itens de maior prioridade do *backlog* são implementados inicialmente [10]. Para definição de quantos itens serão alocados em cada *sprint* foi necessário pontuá-los. A velocidade observada no desenvolvimento determina quantos pontos são passiveis de realização em cada *sprint*. Durante a realização das modificações utilizou-se o conceito de WIP (*work in progress*) para manter foco em atividades e garantir entrega contínua.
Considerando a natureza da assistência técnica prestada, na qual novas atividades eram requisitadas diariamente, e que algumas destas atividades deveriam ser executadas prioritariamente, observou-se o benefício do uso de Scrum no planejamento, pois ao segmentar o trabalho em *sprints* foi mais fácil adaptar a mudanças e garantir a maior entrega de valor ao cliente.
A figura 4 apresenta o resultado do projeto até a semana 16. A linha azul representa a pontuação total dos itens selecionados para execução, a linha vermelha representa o que foi realizado e entregue e a verde corresponde ao atraso. Nas primeiras cinco *Sprints* observa-se um desvio elevado, gerando atraso na entrega. Isso se deveu ao não conhecimento da velocidade real, o que pode ocorrer nos estágios iniciais de um projeto. Inicialmente almejávamos manter o escopo da *Sprint* inalterado ao longo de sua duração, mas a realidade não foi possível. A solução foi reservar tempo para atividades emergenciais e reduzir o número de pontos de cada iteração. Ao analisar o que se conseguiu produzir nas primeiras *Sprints* limitou-se em oitenta pontos como máximo a ser planejado.

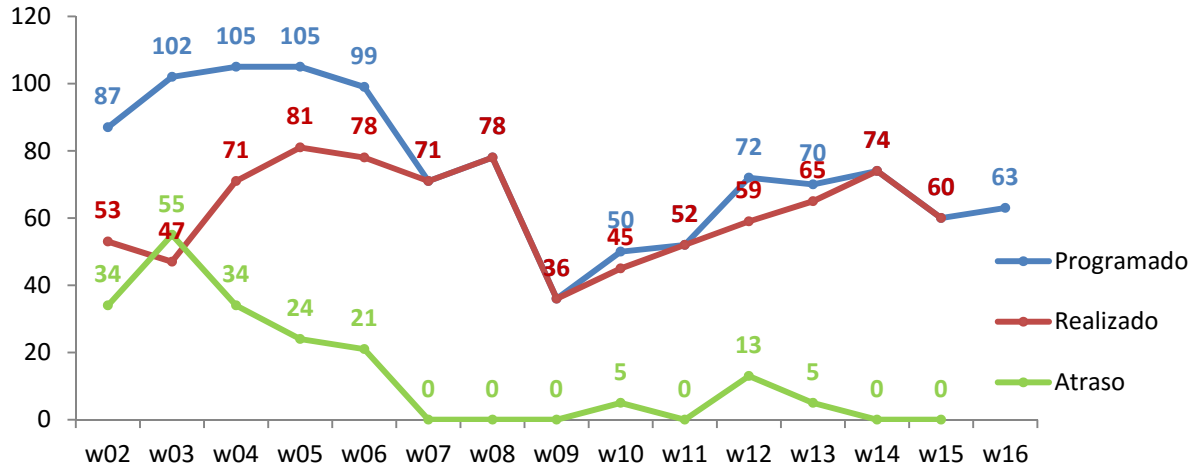

Figura 4. Planejado x realizado dos itens do PB nas *Sprints*.

ISSN: 2237-0234

3 RESULTADOS

O acompanhamento de resultados foi realizado principalmente analisando-se a evolução do TBC, indicador de maior significado para o projeto, pois sua redução significava entregar valor ao cliente. Em adição também foram analisados o grau de utilização do laminador, que contempla um aspecto mais amplo se comparado ao TBC, pois não se restringe a paradas menores que cinco minutos.
A figura 5 é parte de um relatório e apresenta o resultado compilado pelo cliente. A semana 18 representada no gráfico corresponde a *Sprint* 5 deste trabalho. Ao analisarmos a figura observa-se uma melhora constante no decorrer das Sprints como consequência da entrega contínua das modificações.

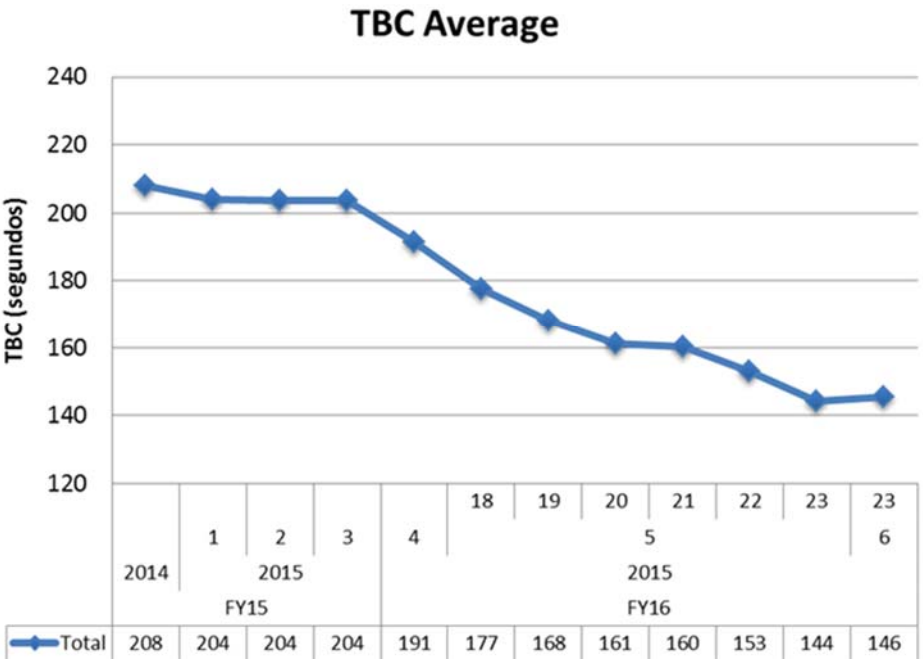

Figura 5. Evolução da média semanal do TBC.

A figura 6 apresenta o resultado do mesmo período, porém em forma de histograma. Nela é possível observar a redução na dispersão das amostras, o que acusa o aumento na confiabilidade do equipamento. Este resultado foi decorrente principalmente das ações voltadas para redução de falhas mais frequentes e do aumento da confiabilidade das lógicas de controle.
A reorganização das mensagens do sistema de alarmes também foi fator importante por ter propiciado ao operador se concentrar primariamente nos aspectos mais importantes, que tem maior poder de afetar a produção. Com o sistema de mensagens poluído com várias informações de baixa prioridade o que de fato ocorria era a desconsideração generalizada.

ISSN: 2237-0234

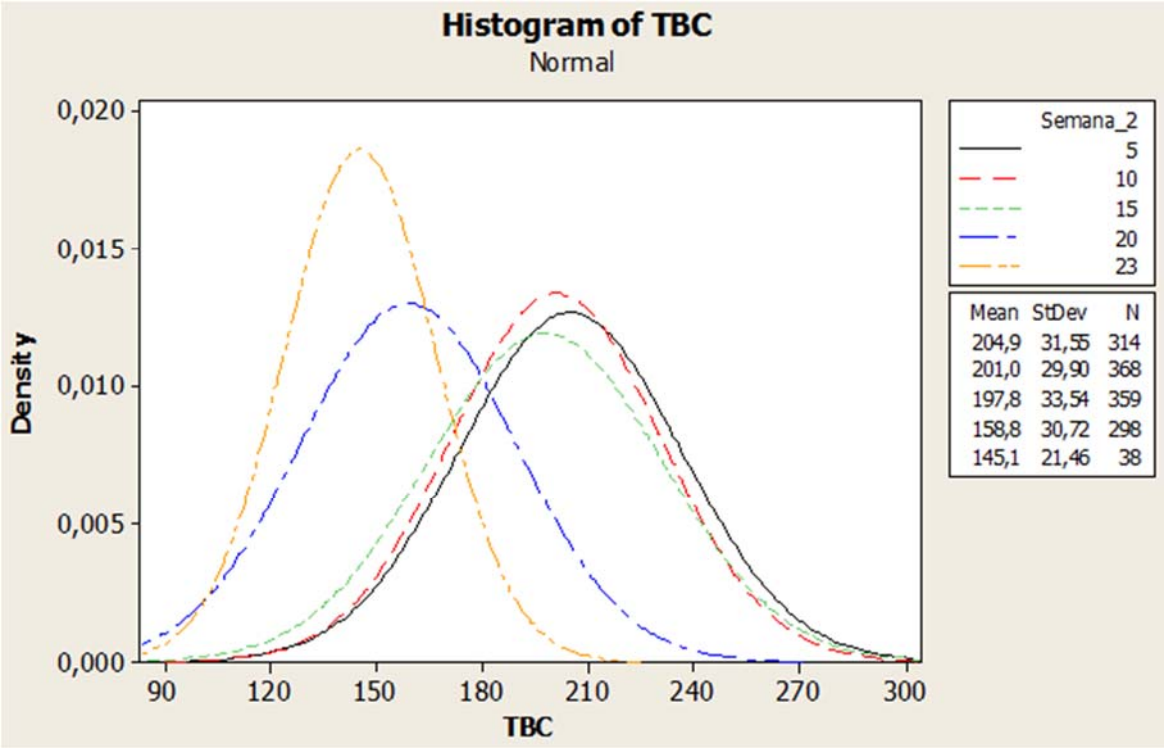

Figura 6. Histograma semanal do TBC.

Dentre as ações para redução do TBC, a realização da tela Kaizen Online em especial foi importante para alcançar o resultado e merece destaque, pois se baseou no conceito de gestão a vista como se realiza nos quadros do Kanban e em outras práticas ágeis.

A figura 7 exibe a parte principal da tela Kaizen Online. O objetivo era que os operadores pudessem acompanhar em tempo real o desempenho de seu turno e realizar comparações com outras turmas. Também era possível comparar a média do mês atual com o realizado no mês anterior. Na parte superior da tela fica a informação do último TBC medido, de forma estratificada, para que seja fácil identificar em qual área do equipamento há falta de desempenho. Em uma segunda iteração foi desenvolvida a parte inferior onde é exibida a média de cada turma para o mês atual e o anterior. Por fim foram desenvolvidas telas para acompanhamento do tempo decorrido de movimentação de cada equipamento. Desta forma a atividade de identificação de anomalias foi simplificada e o diagnostico ficou mais preciso.

Possibilitar que os operadores acompanhassem a evolução do TBC e realizassem comparações entre turmas proporcionou que os mesmos tomassem ações para melhorar o desempenho de sua turma, fato que proporcionou uma maior equalização das médias de cada turma. Com as boas práticas disseminadas o desempenho geral do TBC melhorou mesmo sem novos desenvolvimentos por parte do software. A padronização das práticas de operação também colaborou para redução do desvio padrão nas amostras como observado na figura 6.

ISSN: 2237-0234

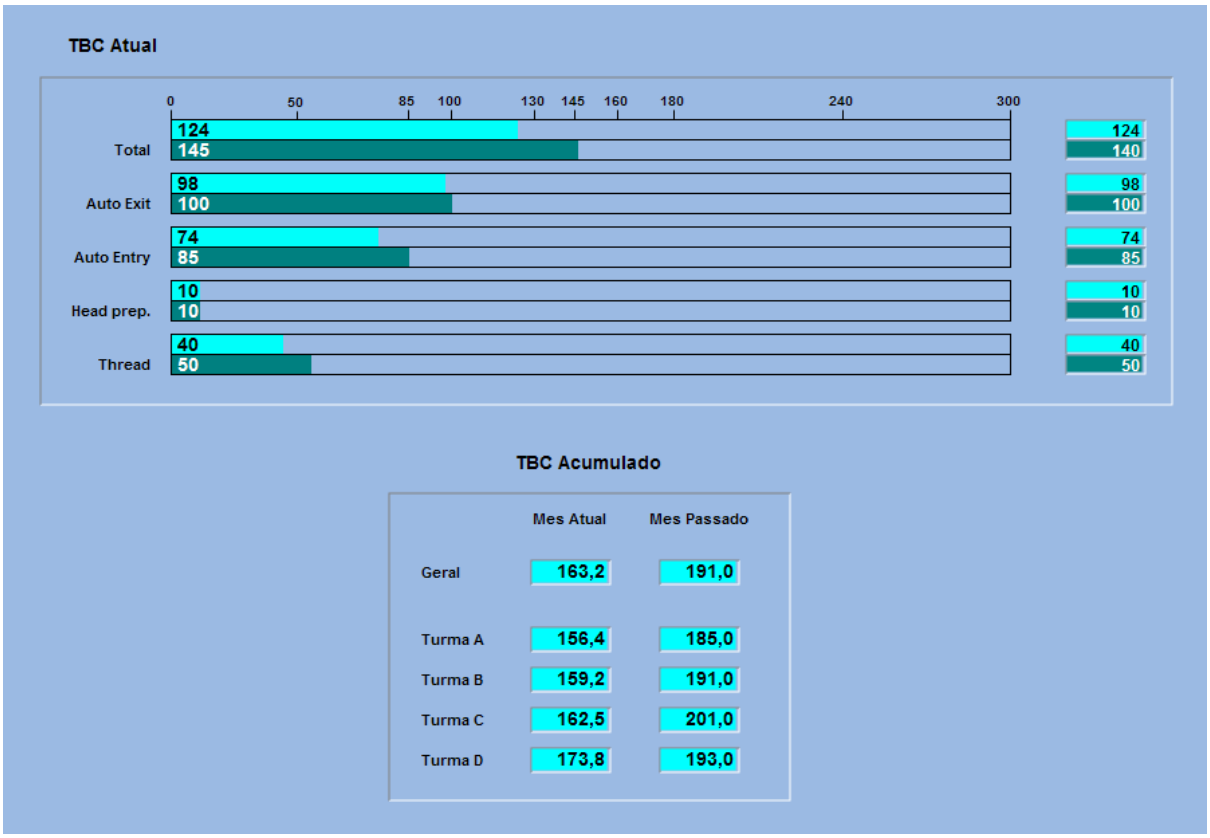

Figura 7. Tela principal do Kaizen Online.

O resultado da realização da tela Kaizen Online foi um aumento considerável da conscientização da operação para a melhoria do indicador. A média nas semanas seguintes a sua disponibilização continuou a cair em decorrência do esforço da operação, mesmo sem a entrega de novas ações.

5 CONCLUSÃO

O objetivo deste trabalho foi apresentar um estudo da utilização de metodologias ágeis no desenvolvimento de software para automação industrial. No caso específico em que este projeto foi aplicado entrega de valor ao cliente consistiu em reduzir o tempo ocioso entre bobinas (TBC) do laminador CM#3 da Novelis através de pequenas alterações e otimizações de lógicas de controle e automação.
Analisando os indicadores do projeto conclui-se que a utilização de técnicas ágeis impactou de forma benéfica o resultado do projeto. É importante lembrar, entretanto que este estudo de caso focou em uma situação específica, dado o contexto da automação industrial e, devido a restrições de projeto e de tempo, aplicou as práticas ágeis de forma superficial. Tanto o campo da automação quanto o de métodos ágeis possuem uma ampla área de conhecimento e seria necessário um esforço maior para obtermos uma visão mais ampla sobre o que foi proposto.
Aplicar métodos ágeis em automação industrial proporcionou grandes desafios. Dentre as maiores dificuldades estão a baixa maturidade e adesão da industria às tendências da tecnologia de informação. A diferença cultural também é um empecilho quando se tenta adaptar a forma de trabalho. Em compensação parte da indústria é familiarizada com técnicas de gestão da produção como *Lean*, *Just in Time* e suas variações. Analisando especificamente os métodos ágeis escolhidos concluímos que aqueles que focam em gerenciamento como Scrum, são mais fáceis

de implementar do que os que afetam o desenvolvimento como TDD. O que pode ser observado neste ponto é uma barreira cultural, natural quando se propõe modificações de comportamento.

Um indicador de que o trabalho foi realizado com sucesso foi a extensão do serviço além do prazo estipulado. Inicialmente foram previstos um mês com possibilidade de prorrogação por mais um. Os resultados do projeto começaram a ser observados a partir da segunda semana. Cientes de que estávamos atingindo os objetivos dentro do prazo, houve incentivo por parte do cliente a traçar novas metas e continuar a evolução do sistema. Ao final a assistência se estendeu por cinco meses.

REFERÊNCIAS

1 V. VYATKIN: "*Software Engineering in Factory and Energy Automation: State of the Art Review*", IEEE Transactions on Industrial Informatics, 2013.
2 R. STETTER.: *"Software im Maschinenbau –lästiges Anhängsel oder Chance zur Marktführerschaft?"* Verband Deutscher Maschinen- und Anlagenbau, 2011.
3 D. VERMA, V. GUPTA: "*Agile Methodologies for different Industries"*, IJREAT International Journal of Research in Engineering & Advanced Technology 2014.
4 G. KUMAR, P. BAHTIA: *"Impact of Agile Methodology on Software Development Process"* International Journal of Computer Technology and Electronics Engineering (IJCTEE) 2012.
5 C. KLOUKINAS, D. ROTONDI et al: "*Agile Manufacturing General Challenges and an IoT@Work Perspective*" Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference 2012.
6 A. BEGEL, N. NAGAPPAN: "*Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study*", First International Symposium on Empirical Software Engineering and Metrics, 2007.
7 K. BECK, M. BEEDLE, A.V. BENNEKUM, A. COCKBURN e W. CUNNINGHAM et al: "*Manifesto Ágil*" (2001). Disponível em: http://agilemanifesto.org/. Acesso em: 9.jul.2015
8 TAGHI JAVDANI G., HAZURA ZULZABIL et al.: "*Obstacles in moving to agile software. Development methods at a glance*" Journal of Computer Science 2013.
9 K. PETERSEN, C. WOHLIN: "*A Comparison of Issues and Advantages in Agile and Incremental Development between State of the Art and an Industrial Case"*, Journal of Systems and Software 2009.
10 SCRUM ALLIANCE Disponível em: https://www.scrumalliance.org/. Acesso em: 8.jul.2015.
11 HIROTAKA T., IKUJIRO N.: *"The New Product Development Game"* Harvard Business Review 1986.
12 EYLEAN: *"Whitepaper - Scrum vs Kanban vs Scrumban"* Disponível em: http://www.vwea.org/storage/documents/Events/EducationSeminar/c2%20%20%20vwea%20presentation%20-%20abh.pdf. Acesso em: 8.jul.2015.
13 KENNETH S. RUBIN: *"Essential Scrum: A Practical Guide to the Most Popular Agile Process"* Primeira edição, Addison-Wesley Professional, Michigan 2012.
14 TERENCE PARR: "Extreme Programming" Disponível em: http://www.cs.usfca.edu/~parrt/course/601/lectures/xp.html Acesso em: 12.jul.2015
15 A. COCKBURN, L. WILLIAMS: *"The Costs and Benefits of Pair Programming"* Proceedings of the First International Conference on Extreme Programming and Flexible Processes in Software Engineering 2000.
16 AGILE ALLIANCE Disponível em: http://agilealliance.org. Acesso em: 10.jul.2015.
17 L. WILLIAMS, E. MICHAEL MAXIMILLIEN: *"Assessing Test-Driven Development at IBM"* Proceedings of the 25th International Conference on Software Engineering 2003.
18 IEC 61131-3: "*IEC 61131-3 Standard - Programmable controllers - Part 3: Programming languages, 2nd ed*". International Electrical Commission, 2003.

19 ISPE GAMP D-A-C-H SIG ASDMM: "*Alternative Software Development models and methods in GxP environments*" The Official magazine of ISPE 2012.
20 ANDREW HIGGINBOTHAN *"Replacing a Collection System's Control System (SCADA) Using Agile Methods"* 2014. Disponível em: http://www.vwea.org/storage/documents/Events/EducationSeminar/c2%20%20%20vwea%20presentation%20-%20abh.pdf. Acesso em: 8.jul.2015.
21 D. WINKLER, S. BIFFL, T. ÖSTREICHER, "*Test-Driven Automation - Adopting Test-First Development to Improve Automation Systems Engineering Processes*," in Proceedings of the 16th EuroSPI Conference, 2009.
22 HAMETNER, R., WINKLER, D. & ZOITL, A.: "*Agile Testing Concepts Based on Keyword-driven Testing for Industrial Automation Systems*" IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, 2012.
23 SIMATIC S7-400: Disponível em: http://w3.siemens.com.br/automation/br/pt/automacao-e-controle/automacao-industrial/simatic-plc/s7-cm/s7-400/Pages/Default.aspx. Acesso em 25 de abril de 2016.
24 B. VOGEL-HEUSER, et al.: "Challenges for Software Engineering in Automation" Journal of Software Engineering and Applications, 2014.