CONTRIBUIÇÃO DOS PENSAMENTOS ORIUNDOS DA MELHORIA CONTÍNUA NA OTIMIZAÇÃO DE SOFTWARES NA MICROSOFT

Resumo

Este artigo aborda de forma analítica, como os pensamentos da melhoria contínua influenciam positivamente a otimização de softwares principalmente na Microsoft, e tem como objetivo analisar as práticas de melhoria contínua adotadas pela empresa em seus processos de desenvolvimento de software. Para alcançar esse objetivo, foi realizada uma pesquisa iniciada em março de 2023 até maio de 2023, baseada em estudos de caso, análise documental e revisão da literatura. O estudo revela os benefícios e resultados alcançados pela Microsoft ao aplicar práticas como o PDCA (*Plan, Do, Check, Act*) em seus processos de desenvolvimento de software, incluindo a identificação de problemas, coleta de dados, implementação de ações corretivas, estímulo à comunicação e colaboração, revisões periódicas do trabalho e padronização de processos. Mostrando assim, como conceitos da melhoria contínua não só contribuem, mas são necessários no desenvolvimento e otimização de software.

Palavras-chave: *Software, Melhoria contínua, Microsoft, Otimização, contribuição pensamento.*

1.Introdução

A melhoria contínua é uma abordagem fundamental para aprimorar constantemente os processos e produtos de uma organização. Ela desempenha um papel crucial no desenvolvimento de *software*s, especialmente no contexto da Microsoft, líder no setor de desenvolvimento de *software*. A Microsoft é conhecida por adotar práticas, *frameworks* e métodos que promovem a otimização contínua de seus *softwares*, o que resulta em produtos de alta qualidade e satisfação dos clientes (Tecfy Bussiness Solutions, 2022).

O objetivo geral desta pesquisa, é investigar como as práticas, *frameworks* e métodos oriundos da melhoria contínua, como o PDCA e o *Scrum*, contribuem para a otimização de softwares na Microsoft.

2.Fundamentação teórica

2.1Princípios fundamentais da melhoria contínua

A melhoria contínua é uma abordagem sistemática para aprimorar processos, produtos e serviços, que tem como objetivo proporcionar um aumento constante de eficiência, qualidade e satisfação para as partes interessadas. Para que isso seja possível, é necessário compreender alguns princípios fundamentais que norteiam a melhoria contínua. Segundo Shingo (1996), melhoria contínua seria um processo que visa alcançar melhorias gradativas, em vez de mudanças radicais e disruptivas. Já para Juran (1980), a melhoria contínua deveria ser uma atividade planejada, sistemática e persistente. Ver quadro 1.

Quadro 1 – princípios fundamentais da melhoria contínua

Princípio	Descrição
Primeiro princípio	O primeiro princípio é o de que tudo pode ser aprimorado. Esse princípio se baseia na ideia de que não existe um estado de perfeição definitivo, mas sim uma busca constante por melhorias. Dessa forma, é necessário ter uma postura proativa e buscar continuamente formas de aprimorar processos, produtos e serviços. A abordagem sistêmica é um dos princípios fundamentais da melhoria contínua, que consiste em considerar todos os aspectos envolvidos no processo de melhoria. Segundo Deming (2000), a abordagem sistêmica leva em conta as interações entre os diferentes elementos do sistema, identificando as causas raiz dos problemas e propondo soluções integradas. Por isso, é importante ter uma visão crítica dos processos existentes, buscando identificar oportunidades de melhorias em cada etapa.
Segunda princíoio	O segundo princípio segundo Deming (2000) é o de que a melhoria contínua deve ser uma responsabilidade compartilhada. Isso significa que todos os membros de uma organização devem estar envolvidos no processo de melhoria contínua, contribuindo com ideias, sugestões e *feedbacks* construtivos. Além disso, é importante que a liderança da organização promova uma cultura de melhoria contínua, incentivando a colaboração e a participação de todos os envolvidos.

Fonte; Deming, 2000

Compreender e aplicar esses princípios é fundamental para o sucesso da melhoria contínua. Dessa forma, é necessário adotar uma abordagem sistêmica que considere os processos como um todo, levando em conta suas interações e impactos. Conforme Liker e Franz (2011), o ciclo PDCA é uma ferramenta que ajuda a identificar as oportunidades de melhoria, definir e implementar soluções, verificar os resultados e atuar para corrigir desvios e manter os processos em conformidade. O ciclo PDCA é uma metodologia de melhoria contínua amplamente

utilizada na gestão de processos. O acrônimo PDCA representa as etapas do ciclo: *Plan* (planejamento), *Do* (execução), *Check* (verificação) e *Act* (ação).

Utilizando o referencial de Grattan-Guinness (2005) no capítulo 72, onde é trabalhado o livro de Walter A. Shewhart que traz à contexto os princípios estatísticos de controle e qualidade, destaca-se a importância dessas etapas do processo de melhoria para um processo. Na fase de planejamento, vemos como são definidos os objetivos e metas de melhoria. Na fase de execução, são realizadas as ações planejadas. Na fase de verificação, observamos a avaliação dos resultados obtidos. E na fase de ação, são tomadas as decisões necessárias para corrigir desvios e implementar melhorias contínuas. No contexto do desenvolvimento de *software*, a aplicação dos princípios da melhoria contínua, da abordagem sistêmica e da metodologia PDCA pode contribuir para a otimização dos processos, aprimorando a qualidade dos produtos e serviços oferecidos pela área de programação e análise e desenvolvimento de sistemas.

2.2 Práticas e métodos da melhoria contínua aplicados ao desenvolvimento de *software*

Na área de programação e desenvolvimento de *softwar*e, a melhoria contínua, se observarmos de uma maneira conceitual, verificando as possibilidades de soluções utilizadas para otimizar sistemas, pode desempenhar um papel essencial para aprimorar os processos de desenvolvimento, manutenção e melhoria de *softwares*. Com a crescente demanda por soluções tecnológicas cada vez mais sofisticadas, seguras e eficientes "os investimentos em TI no Brasil estão em alta e estima-se que o aumento seja de 20% até o fim de 2022" (Tecfy Bussiness Solutions, 2022), é fundamental que as empresas adotem práticas e métodos que permitam a otimização contínua de processos como os que baseiam os *softwares* desenvolvidos.

Com base na metodologia PDCA segundo Shewhart (1980) , constata-se como a melhoria contínua pode ser aplicada ao desenvolvimento de *software*, por ele ser formado por operações e sistemas que utilizam e necessitam de práticas e métodos específicos, e consideram pontos como a análise de processos, identificação de oportunidades de melhoria, estabelecimento de metas e objetivos, planejamento de ações de melhoria, implementação de ações de melhoria, verificação dos resultados das ações de melhoria e padronização dos processos aprimorados. Nesta seção, serão analisadas as principais práticas e métodos que podem ser utilizados para alcançar melhorias contínuas no desenvolvimento de *software*.

2.3Análise de processos

Análise de processos é uma prática fundamental da melhoria contínua, que envolve a identificação de processos ineficientes ou com problemas e a busca por soluções. Conforme

afirmado por Campos (1992), A análise de processos é uma atividade que visa à identificação, modelagem e análise dos processos, com o objetivo de identificar as causas dos problemas existentes e, posteriormente, propor melhorias que possam ser implementadas. Aqui, vemos que não há nada que deve ser feito para modificar o processo, apenas a observação e documentação do fluxo de trabalho, devemos buscar identificar gargalos e ineficiências nas etapas do desenvolvimento de *software*. Com base nessa análise, é possível identificar oportunidades de melhoria.

2.4Identificação de oportunidades de melhoria

A identificação de oportunidades de melhoria é outra etapa importante, que consiste em identificar áreas onde a melhoria pode ser implementada. A identificação de oportunidades de melhoria pode ser feita por meio de diferentes técnicas, como análise SWOT, análise de dados e análise de processos. O objetivo é encontrar áreas que apresentam problemas ou que possam ser melhoradas e, a partir disso, estabelecer objetivos e metas para a melhoria. Uma vez identificadas as oportunidades de melhoria, é necessário estabelecer metas e objetivos para que o programa ou sistema aprimore o processo de desenvolvimento do *software*. Utilizando da linguagem de programação para dar prompts quais o sistema deve seguir à risca. As metas e objetivos devem ser específicos, mensuráveis, alcançáveis, relevantes e temporalmente definidos, a fim de garantir a efetividade das ações de melhoria.

2.5 Estabelecimento de metas e objetivos

Uma vez que as oportunidades de melhoria são identificadas, é preciso estabelecer metas e objetivos claros e mensuráveis. Conforme afirma Liker (2011), as metas e objetivos são fundamentais para o processo de melhoria contínua, pois é necessário definir o que se espera alcançar e como isso será medido. As metas devem ser ambiciosas, mas realistas, e devem ser definidas com base em dados e informações objetivas. O estabelecimento de metas e objetivos é um dos princípios fundamentais da melhoria contínua, por isso, devem ser estabelecidos com base em indicadores de desempenho e devem estar alinhados com a estratégia da organização. No contexto do desenvolvimento de *software*, as metas podem estar relacionadas a prazos, qualidade do software, satisfação do cliente, entre outros.

No que diz respeito ao desenvolvimento de *software*, as metas podem estar relacionadas a diferentes aspectos do processo, como prazos, qualidade do *software* e satisfação do cliente. Além disso, a definição de metas e objetivos também pode contribuir para a criação de um

senso de direção e propósito, ajudando a equipe a se manter motivada e engajada no processo de melhoria contínua.

Uma das ferramentas que podem ser utilizadas para estabelecer metas e objetivos no contexto do desenvolvimento de *software* é o *Balanced Scorecard (BSC).* Segundo Kaplan e Norton (1997), o BSC é um sistema de gestão estratégica que permite alinhar as metas e objetivos da organização com as estratégias adotadas, por meio da definição de indicadores de desempenho em diferentes perspectivas, como financeira, cliente, processos internos e aprendizado e crescimento. Dessa forma, é possível monitorar o progresso em relação aos objetivos estabelecidos e promover ações de melhoria contínua ao longo do tempo.

Assim, o estabelecimento de metas e objetivos claros e mensuráveis é uma prática fundamental da melhoria contínua que pode ser aplicada com sucesso no contexto do desenvolvimento de software, contribuindo para a criação de um ambiente orientado a resultados e promovendo a excelência no processo de desenvolvimento de *software*.

2.6 Planejamento de ações de melhoria

Com as metas e objetivos estabelecidos, é preciso planejar as ações de melhoria. O planejamento de ações de melhoria é outra etapa de destaque, que envolve a definição de atividades específicas para alcançar as metas estabelecidas. Segundo Deming (1980), "O planejamento de ações de melhoria deve levar em consideração todos os aspectos envolvidos no processo, incluindo a identificação de recursos necessários, definição de responsabilidades e cronograma de implementação". Por meio dessa prática, é possível identificar as ações que devem ser tomadas, estabelecer prazos e responsabilidades, e monitorar o progresso das melhorias. O planejamento das ações deve considerar todos os pontos citados, recursos necessários, os responsáveis pela implementação e o cronograma de execução. É importante que as ações de melhoria estejam alinhadas com os objetivos estabelecidos anteriormente, a fim de garantir o sucesso do projeto de melhoria.

2.7 Verificação dos resultados das ações de melhoria.

Após a implementação das ações de melhoria, é fundamental verificar os resultados obtidos. A verificação deve ser feita por meio de indicadores, que permitam avaliar o impacto das ações implementadas. Conforme enfatizado por Shewhart (1980), "a verificação é a chave para garantir que as melhorias tenham sido efetivas". É importante medir e analisar os resultados das ações de melhoria para verificar se as metas foram alcançadas e para identificar novas oportunidades de melhoria.

Destaca-se firmemente que a verificação dos resultados deve ser realizada de forma constante, a fim de garantir a continuidade do processo de melhoria contínua. A verificação dos resultados das ações de melhoria é uma etapa fundamental para garantir que as metas estabelecidas foram alcançadas. Segundo Juran (1980), A verificação dos resultados deve ser feita com base em dados objetivos e mensuráveis, para que possa ser avaliada a eficácia das ações de melhoria implementadas. É importante destacar que, em caso de resultados insatisfatórios, é preciso realizar novas análises e identificar novas oportunidades de melhoria.

2.8 Padronização dos processos aprimorados

Por fim, é necessário padronizar os processos aprimorados, a fim de garantir que as melhorias obtidas sejam mantidas ao longo do tempo. Uma ideia amplamente defendida na literatura de melhoria contínua e gestão de processos segue: "a padronização é essencial para garantir a continuidade das melhorias e a sustentabilidade dos processos aprimorados". Autores como Liker (2011) e Shingo (1996) enfatizam a importância da padronização para manter as melhorias implementadas e garantir a qualidade e eficiência dos processos.

Por meio da padronização, é possível garantir que as melhores práticas sejam aplicadas de forma consistente em todo o processo de desenvolvimento de software, promovendo a uniformidade e a previsibilidade dos resultados. Além disso, a padronização facilita a comunicação e a transferência de conhecimento entre os colaboradores, permitindo que as melhorias sejam disseminadas e perpetuadas na organização. A padronização deve ser feita por meio de procedimentos operacionais, que detalham as atividades a serem realizadas, os responsáveis e as formas de avaliação do processo. Esses procedimentos devem ser documentados e disseminados para todos os envolvidos no processo, a fim de garantir a aderência e a uniformidade das práticas. De acordo com Juran e Godfrey (2014), a documentação dos procedimentos operacionais é importante para padronizar as práticas e para facilitar a compreensão e o aprendizado das atividades pelos colaboradores.

2.9 Fases do ciclo PDCA e como elas podem ser implementadas no processo de otimização de *softwares*.

A metodologia PDCA (*Plan, Do, Check, Act*) é uma abordagem amplamente utilizada em diversas áreas para aprimorar processos e garantir a melhoria contínua. Na área de desenvolvimento de *softwar*e, essa metodologia pode ser implementada com sucesso para otimizar processos e melhorar a qualidade do *software* produzido. O quadro 2, a seguir descreve as fases do PDCA (Ferramentas da qualidade, 2021)

Quadro 2 – Descrição das fases do PDCA

Fases do PDCA	Descrição
1ª Fase	A primeira fase do ciclo PDCA é o planejamento *(Plan)*, que consiste na identificação dos objetivos a serem alcançados e no estabelecimento de um plano de ação para atingi-los. Nessa fase, é importante identificar quais processos precisam ser aprimorados e como isso pode ser feito e envolver todos os membros da equipe de desenvolvimento para garantir que os objetivos estabelecidos sejam claros e compreendidos por todos. De acordo com Chaves *et al*. (2021), a fase de planejamento do ciclo PDCA deve ser cuidadosamente planejada para garantir que as ações a serem tomadas sejam eficazes e eficientes. Por exemplo, pode-se analisar o tempo gasto em cada etapa do desenvolvimento e identificar onde há gargalos e desperdícios, com o objetivo de reduzir o tempo total de desenvolvimento.
2ª Fase	A segunda fase do ciclo PDCA é a execução *(Do)*, que consiste na implementação do plano de ação ou melhoria, planejado e estabelecido na fase anterior, no planejamento. Na execução, é importante que todas as atividades sejam documentadas e que os membros da equipe de desenvolvimento estejam cientes de suas responsabilidades, já que ela envolve a capacitação da equipe e a implementação de ferramentas de melhoria. A fase de execução do ciclo PDCA deve ser realizada com precisão para garantir que o plano de ação seja implementado com sucesso. Um exemplo de ação a ser implementada é a utilização de metodologias ágeis, que têm como objetivo aumentar a eficiência do processo de desenvolvimento de *software* e garantir que as funcionalidades entregues estejam alinhadas com as necessidades do cliente.
3ª Fase	A terceira fase do ciclo PDCA é a verificação *(Check)*, que consiste na avaliação dos resultados obtidos na fase de execução, avaliando se as ações de melhorias implementadas foram efetivas. Nessa fase, é importante a coleta de dados e análise de resultados, para que as metas estabelecidas na fase de planejamento sejam avaliadas e comparadas com os produtos obtidos. De acordo com Shewhart (1980), a fase de verificação do ciclo PDCA deve ser realizada com cuidado para garantir que os resultados obtidos sejam confiáveis e representativos. Aqui, pode-se avaliar se houve redução de defeitos no *software* entregue ou se o tempo total de desenvolvimento foi reduzido
4ª Fase	A quarta e última fase do ciclo PDCA é a ação *(Act)*, que consiste na implementação de ações corretivas e na padronização dos processos aprimorados, corrigindo eventuais problemas identificados na fase de verificação e implementando melhorias adicionais. Nessa fase, é importante documentar as ações corretivas implementadas e revisar os processos aprimorados para garantir que sejam padronizados e replicáveis. De acordo com Shewhart (1980), a fase de ação do ciclo PDCA deve ser realizada de forma sistemática, monitorando os resultados e realizando ajustes necessários para garantir a sustentabilidade das melhorias implementadas. No desenvolvimento de *softwares*, é importante a atenção para que os processos dos programas estejam sempre aprimorados, e "atualizados" com as necessidades dos clientes e do mercado.

Fonte: Ferramentas da qualidade. 2021

O PDCA é amplamente utilizado em diversas áreas, incluindo o desenvolvimento de *software*, para promover a melhoria contínua. Na Microsoft, a aplicação dos princípios do PDCA se faz presente em diversas práticas e frameworks utilizados no desenvolvimento de *softwares*, visando a otimização dos processos e a entrega de produtos de alta qualidade.

Assim, a aplicação do ciclo PDCA permite que as empresas possam identificar e corrigir problemas de forma mais eficiente, aumentando a qualidade do *software* e reduzindo os custos de desenvolvimento. Isso torna as empresas mais competitivas no mercado de *software*,

permitindo que elas atendam às demandas dos clientes e se adaptem rapidamente às mudanças no mercado.

2.10 Práticas e *Frameworks* na Microsoft

A Microsoft é reconhecida por adotar práticas e *frameworks* que promovem a melhoria contínua no desenvolvimento de *softwares*. Alguns exemplos relevantes são:

User Research e Feedback: A Microsoft valoriza a coleta de *feedback* dos usuários para orientar o desenvolvimento de seus *softwares*. Por meio de práticas de pesquisa de usuários, testes de usabilidade e análise de *feedback*, a empresa busca compreender as necessidades dos usuários e incorporar melhorias contínuas em seus produtos (Microsoft, 2022).

Revisões Periódicas e Aprendizados: A Microsoft realiza revisões periódicas de seus processos de desenvolvimento, permitindo a identificação de áreas de melhoria e a incorporação de aprendizados em futuros projetos. Essa abordagem facilita a adaptação contínua dos processos e a aplicação dos princípios do PDCA (Microsoft, 2022).

Scrum: O *Scrum* é um *framework* ágil amplamente utilizado na Microsoft e em outros contextos de desenvolvimento de *software*. Ele promove a colaboração, transparência e entrega incremental de *software*. Por meio de iterações chamadas de *sprints*, o *Scrum* permite uma abordagem adaptativa e contínua, alinhada com os princípios da melhoria contínua. (Schwaber & Sutherland 2020).

React para Windows: O *React* para Windows é um conjunto de bibliotecas e ferramentas desenvolvidas pela Microsoft que permite o uso do *framework React* no desenvolvimento de aplicativos Windows. Ele oferece suporte ao desenvolvimento de interfaces de usuário modernas e reativas, utilizando componentes reutilizáveis e uma abordagem baseada em componentes. Essa abordagem facilita a inspeção e melhoria contínua das interfaces de usuário desenvolvidas, permitindo aperfeiçoamentos constantes (Microsoft, 2022).

3Metodologia

Com uma pesquisa de natureza exploratória, Ganga, (2012), afirma que busca expandir a dimensão da visibilidade do tema abordado, na tentativa de fornecer uma familiaridade maior aos pontos que relacionam o uso de técnicas de melhoria contínua como o PDCA em *softwares*. A Microsoft será selecionada como estudo de caso devido à sua relevância e reconhecimento na indústria de *softwar*e. Foi realizada uma análise de natureza documental, portanto majoritariamente secundária, para compreender como a empresa utiliza os pensamentos oriundos da melhoria contínua em seus processos de desenvolvimento de *software.*

Destaca-se com Deming (1980) como que a melhoria contínua é uma filosofia que busca aperfeiçoar continuamente um processo, produto ou serviço, visando sempre a excelência e a satisfação do cliente. Através dela, é possível identificar oportunidades de melhoria, planejar e implementar ações corretivas e preventivas, monitorar e avaliar resultados, e agir de forma a manter o processo sempre em evolução, dessa forma é evidente que a evolução do digital, da *web* e de *softwares* se encaixam perfeitamente nesse contexto de continuamente estar em processo de melhoria.

Para identificar as práticas e os métodos da melhoria contínua que podem ser aplicados no desenvolvimento de *software*, é importante destacar que a área de tecnologia da informação também se beneficia desses princípios. Conforme proposto por Wohlin (2012), a aplicação de metodologias ágeis, como o *Scrum*, pode ser vista como uma forma de melhoria contínua, uma vez que busca aprimorar constantemente o processo de desenvolvimento de *software* e a entrega de valor ao cliente. Com base na pesquisa bibliográfica, descreveu-se a análise de dados disponíveis estão correlacionados com os princípios que sustentam o *Kaizen*.

4Análise dos resultados e discussões

4.1Benefícios e resultados alcançados

Os resultados indicam que a empresa Microsoft, no segmento de desenvolvimento de *softwares*, busca sempre aprimorar suas práticas e seus métodos, adaptando conceitos e *frameworks* que incentivam uma melhoria contínua, também promovida pelo *Kaizen*. Como base na gestão, observou-se como se tornou notório o desenvolvimento desse setor da empresa, garantindo sua autoridade no assunto, com um leque de ofertas que oferece diversos programas que impulsionam a necessidade do consumidor de crescer e melhorar, bem como a evolução do segmento como um todo.

Observa-se que os clientes ficaram mais receptivos aos seus programas e conceitos, devido ao *feedback* constante, através de testes e garantias de qualidade, *user research*, mantendo uma entrega contínua de melhoria. Está enraizado nas suas práticas os conceitos, estabelecido em sua documentação os métodos, e está incentivado no seu processo, os *frameworks* de melhoria contínua.

Algumas das melhorias que a Microsoft alcançou ao aplicar práticas de melhoria contínua, como o PDCA, em seus processos de desenvolvimento de *software* que também podem ser destacadas no quadro 3, a seguir.

Quadro 3 - melhorias que a Microsoft alcançou ao aplicar práticas de melhoria contínua

Melhoria	Descrição
Aumento da qualidade do software	Através da identificação proativa de problemas e oportunidades de melhoria, a Microsoft conseguiu aprimorar a qualidade de seus *softwares*. Isso resultou em menos *bugs*, maior estabilidade, melhor desempenho e maior satisfação do cliente.
Ciclos de desenvolvimento mais eficientes	Com a implementação de ações corretivas e de melhoria em ciclos contínuos, a Microsoft conseguiu acelerar o processo de desenvolvimento de *software*. Isso permitiu a entrega mais rápida de novas funcionalidades e atualizações aos usuários, mantendo um ritmo constante de evolução.
Maior satisfação do cliente	Ao ouvir ativamente os usuários, coletar *feedback* e incorporar essas informações nos processos de melhoria contínua, a Microsoft conseguiu aprimorar a experiência do cliente. Isso resultou em maior satisfação dos usuários e aumento da fidelidade à marca.
Redução de retrabalho	Com a detecção precoce de problemas e a implementação de ações corretivas, a Microsoft conseguiu reduzir a ocorrência de retrabalho. Isso economiza tempo e recursos, permitindo que a equipe se concentre em tarefas mais produtivas e no desenvolvimento de novas funcionalidades.
Maior eficiência operacional	A padronização de processos e a definição de boas práticas na entrega contínua permitiram à Microsoft melhorar a eficiência operacional. Os fluxos de trabalho automatizados e consistentes simplificaram as etapas de desenvolvimento, teste e implantação, resultando em um fluxo de trabalho mais suave e eficiente.
Melhoria contínua da equipe	Através das revisões periódicas do trabalho e da colaboração entre os membros da equipe, a Microsoft promoveu uma cultura de aprendizado e melhoria contínua. Os desenvolvedores puderam aprimorar suas habilidades, compartilhar conhecimentos e crescer profissionalmente, contribuindo para um ambiente de trabalho mais motivador e produtivo.

Fonte: Autor, 2023

Essas são apenas algumas das melhorias alcançadas pela Microsoft ao adotar práticas de melhoria contínua. Cada empresa pode ter resultados específicos, mas a abordagem do PDCA geralmente leva a um aprimoramento contínuo em todos os aspectos do desenvolvimento de *software*.

4.2 Metodologia *agile* da Microsoft

A Microsoft adotou a metodologia Ágil em seus processos de desenvolvimento de *software*, com o objetivo de entregar produtos de alta qualidade, aumentar a eficiência e responder rapidamente às necessidades dos clientes. O principal *framework* ágil utilizado pela Microsoft é o *Scrum*, um *framework* ágil para gerenciamento de projetos. Ele é baseado em equipes auto-organizadas, colaboração intensiva e inspeção e adaptação constantes. O *Scrum* visa maximizar a eficiência e a eficácia do desenvolvimento de *software*, permitindo que as equipes respondam de forma flexível às mudanças e entreguem valor aos clientes de maneira mais rápida.

Numa publicação apresentada por desenvolvedores de *software* da Microsoft , Jacobs, Kirch e Kaim, (2023) "O que é *Agile*?", os autores expõem, com outras palavras, a influência da abordagem PDCA no pensamento e estruturas que se estabeleceram os termos e práticas do

método *Agile*, sendo necessário um planejamento, depois é preciso se ter desenvolvimento e entrega de produto, para poder checar e agir sob os resultados ativamente nas operações. O quadro 4, a seguir, apresenta os principais papéis de uma equipe *Scrum*.

Quadro 4 – Principais papéis de uma equipe *Scrum*.

Papéis da equipe *Scrum*	**Descrição**
O *Product Owner*	É responsável por representar os interesses dos clientes, definir as necessidades do produto e priorizar o trabalho no *backlog* do produto. O *Product Owner* mantém um *backlog* do produto, uma lista priorizada de funcionalidades, correções e melhorias necessárias para o produto.
O *Scrum Master*	é o facilitador do processo, garantindo que a equipe *Scrum* siga as práticas e removendo quaisquer obstáculos que possam surgir.
Time de Desenvolvimento	O Time de Desenvolvimento é responsável por realizar o trabalho necessário para entregar incrementos de software no final de cada sprint

Fonte: Autor, 2023

Antes de cada *sprint*, o Time de Desenvolvimento, o *Scrum Master* e o *Product Owner* realizam uma reunião de planejamento de *sprint* para selecionar as funcionalidades a serem desenvolvidas durante o sprint.

Um sprint é um período fixo de tempo, durante o qual o Time de Desenvolvimento trabalha para entregar uma melhoria de *software*. Durante o *sprint*, a equipe realiza reuniões diárias de acompanhamento de progresso, onde cada membro compartilha o que fez, o que fará e quaisquer obstáculos encontrados. No final de cada *sprint*, há duas principais cerimônias: a revisão de *sprint*, onde o Time de Desenvolvimento apresenta o trabalho concluído, e a retrospectiva de *sprint*, onde a equipe analisa o processo e identifica melhorias para os próximos *sprints.*

4.3 Aplicação da melhoria contínua na Microsoft:

A relação entre *Scrum* e *Kaizen* (termo japonês para "melhoria contínua") reside na busca por melhoria constante. Ambos os conceitos compartilham o foco em inspeção e adaptação para aprimorar continuamente os resultados. O *Scrum* promove a melhoria contínua ao incentivar as equipes a refletirem sobre o processo de desenvolvimento de *software* em cada retrospectiva de *sprint*. Durante essa cerimônia, a equipe identifica pontos fortes, pontos fracos e oportunidades de melhoria, que são então implementadas no próximo *sprint.* O *Kaizen*, por sua vez, é um princípio de melhoria contínua em diversos contextos, incluindo processos de produção e

gestão. Ele enfatiza a busca constante por melhorias incrementais, a participação de todos os envolvidos e a mentalidade de aprendizado. Portanto, a relação entre *Scrum* e *Kaizen* está na abordagem compartilhada de promover melhorias contínuas, seja no desenvolvimento de software com *Scrum* ou em outros contextos organizacionais com o *Kaizen.* Ambos os conceitos incentivam a reflexão, a adaptação e a busca por aprimoramento.

A Microsoft promove a inspeção regular dos processos e produtos, a retroalimentação constante dos clientes e a adaptação contínua para otimizar seus *softwares* através de testes de qualidade rigorosos conduzidos em todas as etapas de desenvolvimento como evidenciados no artigo "Noções Básicas de Testes de Unidade" (Microsoft, 2022). Isso inclui, portanto, testes automatizados, testes de unidade, testes de integração e testes de aceitação para garantir que o *software* esteja funcionando corretamente.

Também são realizadas pesquisas com os usuários para entender suas necessidades e expectativas, e pôr em ação as atitudes mais apropriadas para obter o produto mais otimizado. Isso inclui entrevistas, estudos de usabilidade, grupos de foco e análise de dados de uso. As práticas de pesquisa de usuário da Microsoft são mencionadas em artigos e postagens de blog disponíveis no site oficial da Microsoft. Com base nesses *insights,* a empresa obtém feedback valioso que direciona melhorias contínuas no *software.*

Um ponto que solidifica a participação dos conceitos de melhoria contínua pela Microsoft aparecem pela abordagem de lançamento contínuo dos seus programas, quando são fornecidas atualizações regulares para seus *softwares*. Essas atualizações podem incluir correções de *bugs*, melhorias de desempenho, novos recursos e funcionalidades solicitadas pelos clientes; pontos obtidos na análise periódica dos dados qualitativos e quantitativos como tempo de resposta, taxa de erros e uso de recursos oriundos das necessidades dos usuários e dos próprios *softwares* para melhor adaptação às reais necessidades.

5.Considerações finais

Investigou-se neste estudo como as práticas, *frameworks* e métodos oriundos da melhoria contínua, como o PDCA e o *Scrum*, contribuem para a otimização de softwares na Microsoft podendo ser uma referência valiosa para outras empresas. Ao implementar práticas como o PDCA (*Plan-Do-Check-Act*), a Microsoft conseguiu obter diversos benefícios significativos em seus processos de desenvolvimento de *software*. Devido ao baixo índice de abordagem deste tópico, e levando em consideração a grande importância da internet, e dos *softwares* que a compõem, no futuro da sociedade, bem como o quanto a melhoria contínua afetou

positivamente vários tipos de processos produtivos, entende-se como importante, abordar teoricamente o estudo evidenciado nesta tese.

Através da análise dos princípios fundamentais da melhoria contínua, baseados nas contribuições de autores renomados como William E. Deming e Juran, evidenciou-se a importância desses princípios na otimização de *softwares*. Ao examinar as práticas, métodos e *frameworks* utilizados pela Microsoft, como Testes de Qualidade, *User Research*, *Feedback*, *Scrum*, *Kanban*, *Lean Software Development*, Azure DevOps, Microsoft.NET e *React* Windows, constatamos a aplicação concreta da melhoria contínua em seu processo de desenvolvimento.

No que diz respeito às diferentes fases do ciclo PDCA, observou-se uma relação direta entre essas fases e os métodos e processos adotados pela Microsoft na otimização de *softwares*. A satisfação do cliente também foi aprimorada graças à melhoria contínua, pois a Microsoft ouviu ativamente seus usuários, coletou *feedback* e implementou melhorias com base nesses *insights*. Isso resultou em uma experiência do cliente aprimorada e maior fidelidade à marca. Fica claro a grande viabilidade do estudo realizado, pelo fácil acesso à documentos da Microsoft, que mostram abertamente seus *softwares*, *frameworks* e processos, além da bibliografia referente aos princípios *Kaizen* que também pode ser acessada para futuros estudos.

É evidente que a aplicação da metodologia PDCA proporcionou diversos benefícios significativos para a Microsoft. Houve um aumento notável na qualidade dos *softwares*, resultando em menos bugs, maior estabilidade, melhor desempenho e maior satisfação do cliente. A implementação de ciclos de desenvolvimento mais eficientes permitiu uma entrega mais rápida de novas funcionalidades e atualizações, mantendo um ritmo constante de evolução. A escuta ativa dos usuários, a coleta de *feedback* e a incorporação dessas informações nos processos de melhoria contínua resultaram em uma maior satisfação do cliente e maior fidelidade à marca.

A Microsoft alcançou uma redução significativa do retrabalho, economizando tempo e recursos preciosos. A padronização de processos, definição de boas práticas e implementação de fluxos de trabalho automatizados contribuíram para uma maior eficiência operacional. Por fim, a melhoria contínua também teve impacto na equipe, promovendo um ambiente de trabalho motivador e produtivo, onde ocorreu um constante aprimoramento das habilidades e um crescimento profissional contínuo.

Esses resultados evidenciam que a abordagem adotada pela Microsoft, com base na melhoria contínua e no ciclo PDCA, pode servir como referência para outras empresas que desejam otimizar seus processos de desenvolvimento de *software*. Ao implementar práticas de melhoria contínua como o PDCA e enfatizar a qualidade do processo, satisfação do cliente, eficiência operacional e desenvolvimento da equipe, é possível alcançar melhorias significativas nos produtos e processos, obtendo uma vantagem competitiva no mercado em constante evolução. Por isso existe uma importância de aumentar as correlações e testar as aplicações mais diretas, de ferramentas de melhoria contínua do desenvolvimento de *softwares*.

Referências

BAYART, D. "**Economic control of quality of manufactured product" in Grattan-Guinness**. Landmark Writings in Western Mathematics. Elsevier: 926–935,2005.

CAMPOS, V.F. **T.Q.C - Controle da Qualidade Total (no estilo japonês)**. Belo Horizonte: Fundação Christiano Ottoni. Escola de Engenharia, 1992.

DEMING, W E. **Out of the Crisis**. Cambridge, MA: MIT Press, 1982

ENDEAVOR. **Kaizen: a técnica que vai ajudar a alavancar sua empresa.** Endeavor Brasil. Disponível em: https://endeavor.org.br/operacoes/kaizen/. Acesso em: 27 maio 2023.

FERRAMENTAS DA QUALIDADE. PDCA. **Ferramentas da Qualidade.** 2021 Disponível em: https://ferramentasdaqualidade.org/pdca/. Acesso em: 06 jun. 2023.

GANGA, G.M.D. **Trabalho de conclusão de curso**: um guia prático de conceito e forma. São Paulo: Atlas, 2012

JACOBS, M; KIRCH, J; KAIM, E. O que é Ágile. Disponível em: https://learn.microsoft.com/pt-br/devops/plan/what-is-agile. Acesso em: 25 maio 2023.

JURAN, J. M.**Quality Handbook: The Guide to Performance Excellence**, New York, New York: McGraw-Hill, 1974.

KAPLAN, R. S.; NORTON, D. P. **A estratégia em ação: Balanced Scorecard.** Rio de Janeiro: Campus, 1997.

LIKER, J.K. and FRANZ, J.K. **The Toyota Way to Continuous Improvement; Linking Strategy and Operational Excellence to Achieve Superior Performance.** New York, NY. McGraw-Hill, 2011.

LIKER J.K. **The Toyota Way, 14 Management Principles from the World's Greatest Manufacturer**. McGraw-Hill, New York. 2004

LINHA DE CÓDIGO. **Conheça o Microsoft Solutions Framework (MSF).** Linha de Código. Disponível em: http://www.linhadecodigo.com.br/artigo/78/conheca-o-microsoft-solutions-framework-msf.aspx. Acesso em: 01 jun. 2023.

MICROSOFT. **Azure DevOps**. Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/azure/devops/?view=azure-devops-2022. Acesso em: 10 maio 2023.

MICROSOFT. **How Microsoft develops with DevOps.** Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/devops/develop/how-microsoft-develops-devops. Acesso em: 03 jun. 2023.

MICROSOFT. **Microsoft Learn. Microsoft**. Disponível em: https://learn.microsoft.com/pt-br/ Acesso em: 05 maio. 2023.

MICROSOFT. **Microsoft Open Source.** Microsoft. Disponível em: https://opensource.microsoft.com// Acesso em: 05 Mai 2023.

MICROSOFT. **Microsoft Teams.** Microsoft. Disponível em: https://www.microsoft.com/pt-br/microsoft-teams/group-chat-software. Acesso em: 06 jun. 2023.

MICROSOFT. **MSDN Platforms.** Microsoft. Disponível em: https://visualstudio.microsoft.com/pt-br/msdn-platforms/. Acesso em: 04 jun. 2023.

MICROSOFT. ***Periodic Customer Feedback*.** Microsoft Dynamics 365 Release Plan. Disponível em: https://learn.microsoft.com/en-us/dynamics365-release-plan/2020wave1/dynamics365-customer-voice/periodic-customer-feedback. Acesso em: 05 jun. 2023.

MICROSOFT. **Process automation for quality orders.** Microsoft Dynamics 365 Supply Chain Management. Disponível em: https://learn.microsoft.com/pt-br/dynamics365/supply-chain/production-control/process-automation-quality-orders. Acesso em: 02 jun. 2023.

MICROSOFT. **Quality tests**. Microsoft Dynamics 365 Supply Chain Management. Disponível em: https://learn.microsoft.com/pt-br/dynamics365/supply-chain/inventory/quality-tests. Acesso em: 04 jun. 2023.

MICROSOFT. **React overview.** Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/windows/dev-environment/javascript/react-overview. Acesso em: 01 jun. 2023.

MICROSOFT. **Routes and operations.** Microsoft Dynamics 365 Supply Chain Management. Disponível em: https://learn.microsoft.com/pt-br/dynamics365/supply-chain/production-control/routes-operations. Acesso em: 02 jun. 2023.

MICROSOFT. **Saiba mais sobre as melhorias de pesquisa e serviço no Microsoft Edge.** Microsoft Support. Disponível em: https://support.microsoft.com/pt-br/topic/saber-mais-sobre-as-melhorias-de-pesquisa-e-servi%C3%A7o-no-microsoft-edge-5b95a197-6311-4976-ad33-4aad57a9ce65. Acesso em: 05 jun. 2023.

MICROSOFT. **Software License Terms.** Microsoft Legal. Disponível em: https://learn.microsoft.com/en-us/legal/information-protection/software-license-terms-pt-br. Acesso em: 04 jun. 2023.

MICROSOFT. **Troubleshooting.** Microsoft Dynamics 365 Supply Chain Management. Disponível em: https://learn.microsoft.com/pt-br/dynamics365/supply-chain/supply-chain-management-troubleshooting. Acesso em: 02 jun. 2023.

MICROSOFT. **User Research - FAQ**. Disponível em: https://www.microsoft.com/en-us/usability/faq.aspx. Acesso em: 10 maio. 2023.

MICROSOFT. **What is Agile development?** Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/devops/plan/what-is-agile-development. Acesso em: 05 maio 2023.

MICROSOFT. **What is Continuous Integration?** Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/devops/develop/what-is-continuous-integration. Acesso em: 02 jun. 2023.

MICROSOFT. **What is Kanban?** Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/devops/plan/what-is-kanban. Acesso em: 03 jun. 2023.

MICROSOFT. **What is monitoring?** Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/devops/operate/what-is-monitoring. Acesso em: 05 jun. 2023.

MICROSOFT. **What is Scrum?** Microsoft Learn. Disponível em: https://learn.microsoft.com/pt-br/devops/plan/what-is-scrum. Acesso em: 05 jun. 2023.

Schwaber, K., Sutherland, J. **O Guia Scrum. 2020**. Disponível em: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf. Acesso em: em 03 jun. 2023

SHINGO, S. **O sistema Toyota de Produção: Do ponto de vista da engenharia de produção.** Porto alegre, Bookman, 1996

TECFY, B. S. **"Como as empresas se destacam na crescente demanda por soluções tecnológicas?"** Revista Nacional de Tecnologia da Informação, 2022. Disponível em: https://revistati.com.br/noticias/como-as-empresas-se-destacam-na-crescente-demanda-por-solucoes-tecnologicas. Acesso em: 5 jun. 2023.