



GRUPO DE ESTUDO DE SISTEMAS DE INFORMAÇÃO E TELECOMUNICAÇÃO PARA SISTEMAS ELÉTRICOS - GTL

METODOLOGIA ÁGIL E INTEGRAÇÃO CONTÍNUA EM SISTEMAS SCADA EMS

**RAFAEL FRANCO CORDEIRO; FABIO AUGUSTO DA SILVA ANTUNES; CAROLINA CÂNDIDA DE OLIVEIRA
CEMIG GERACAO E TRANSMISSAO S.A**

RESUMO

O trabalho apresenta a evolução do processo de manutenção das bases de dados que compõem o sistema EMS da Cemig GT. Será apresentado como era o processo antes, toda a prospecção de conhecimentos acadêmicos e boas práticas utilizadas no mundo de TI de metodologias ágeis como Kanban, Lean e Devops, a adaptação destes conceitos para a realidade do processo, e a criação de mecanismos de automação, monitoração e acompanhamento do fluxo, com requisitos de validação e auditoria. Na conclusão será apresentado um comparativo entre a metodologia antiga e a nova, evidenciando os ganhos no novo processo.

PALAVRAS-CHAVE

SCADA, EMS, Kanban, Devops, Lean Manufacturing, Integração de Ativos, Operação Remota do SEP

1.0 INTRODUÇÃO

A operação remota de usinas e subestações visa supervisionar diversos parâmetros elétricos relacionados a equipamentos e realizar manobras à distância, de maneira centralizada. A operação remota é amplamente adotada no mercado como forma de reduzir custos operacionais, garantindo a disponibilidade destas estações.

Para que isso ocorra, a manutenção de bases de dados dos sistemas SCADA e EMS deve ocorrer sempre que houver ampliações e alterações. Na CEMIG GT esta atividade ocorre desde 1996 pela equipe do IMSC (Integração e Manutenção do Sistema de Controle) do COS (Centro de Operação do Sistema) e ao longo dos anos passou por constantes evoluções.

O fluxo do processo de atualização de bases de dados se inicia na concepção do projeto da expansão, reforço ou melhoria e passa por diversas etapas até chegar para a equipe do IMSC para o tratamento e implementação da base no sistema SCADA EMS para, então, ser disponibilizada para a operação e, assim, liberar a operação remota daquele ativo.

Historicamente, este processo não expressava uma demanda tão grande de trabalho, porém, com o advento de estratégias ousadas de implantação de reforços e melhorias da companhia (conhecido internamente como "Plano de Obras da Transmissão"), este processo passou a acarretar um grande estresse para as equipes envolvidas.

Neste Plano de Obras de 8 anos, o investimento anual já é 5 vezes maior do que o ano em que houve o maior investimento até então na Cemig e ainda não chegou ao pico de investimentos. Este esforço terá um grande resultado financeiro na RAP da transmissora.

Tendo o motivador de se aumentar a eficiência do processo, atendendo ao Plano de Obras atual e futuro e aproveitando o momento de substituição do sistema de supervisão e controle atual (SSCD) pelo SAGE do Cepel, buscou-se embasamentos teóricos e melhores práticas mundiais com o objetivo de fazer com que o novo processo de atualização de bases de dados fosse mais eficiente, com menos erros, executado por uma equipe mais multidisciplinar e com etapas claras para monitoramento do estado do trabalho de cada escopo a cada momento.

O trabalho envolve o mapeamento de todo o processo, desde a concepção do projeto até a entrada em operação do ativo. Porém, neste primeiro momento, foi destacada a etapa de tratamento de bases de dados no IMSC como a primeira etapa a evoluir, pois notou-se que haveria um maior impacto no valor entregue, com aumento da eficiência do fluxo de trabalho como um todo.

Neste trabalho é apresentado o fluxo antigo, passando por um embasamento teórico e boas práticas, até chegar na descrição do novo fluxo desenhado. Ao final, é apresentado um comparativo entre os fluxos de trabalho e as conclusões, além de indicar possibilidades de trabalhos futuros em favor da evolução contínua do processo, a partir das diretrizes traçadas.

2.0 PROCESSO ANTIGO: MÁSTER

A inclusão de novas bases de dados no COS era executada pelo IMSC através de um processo denominado **Máster**. Definiam-se as datas das másters para o ano, com intervalo de 3 a 4 meses entre elas, para que fosse possível desenvolver todo o processo pela equipe para cada escopo demandado.

Os responsáveis pela automação de usinas e subestações preparavam um documento contendo informações referentes às alterações e ampliações previstas para as estações. Estas informações deveriam ser recebidas pela equipe do IMSC com um prazo de 60 dias antes da data prevista para a máster, para que o escopo pudesse ser devidamente atendido. Na figura abaixo é possível observar que um escopo hipotético “A” poderia ter uma janela de até 180 dias contados entre a entrega das informações de campo até a energização do ativo.

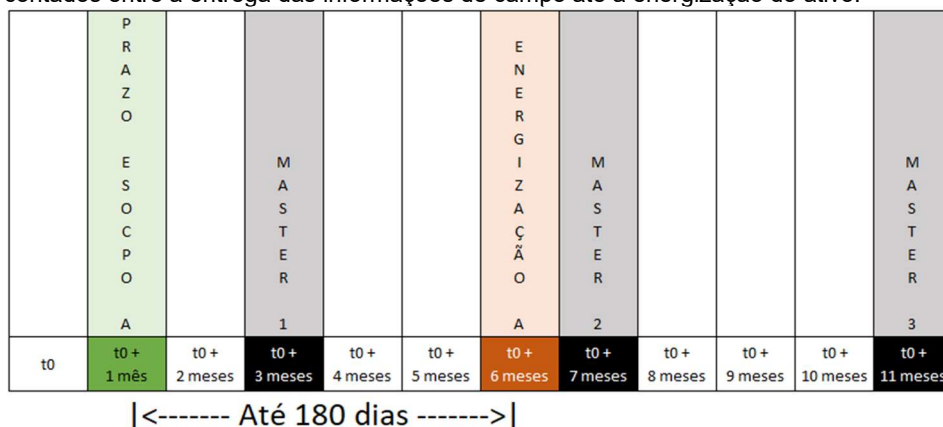


FIGURA 1

No fluxo de trabalho da equipe do IMSC, assim que recebidos os documentos do campo, estes eram repassados para a equipe responsável pelas definições dos requisitos de proteção do SEP para conferência e definição dos modelos de proteção e dos padrões de nomes.

Após a devolução dos documentos com as informações do campo mais as definições de proteção, um grupo interno do IMSC analisava cada base de dados e as traduziam para planilhas Excel, que seriam posteriormente entrada para os diversos módulos do sistema do SSCD como o SCADA (aquisição e distribuição de Pontos Digitais, Pontos Analógicos, Controles, etc.), IHM, Análise de Redes, Controle Automático de Tensão - CAT, Sistema Automático de Preparação de Estações - SAPRE, dentre outros.

Cada módulo recebia estas informações de entrada e equipes especializadas de cada módulo executava os pedidos de serviço que eram implementados em base de dados de desenvolvimento por outra equipe (equipe BD) responsável pelos diversos bancos de dados existentes.

Ao final da incorporação das modificações executadas pela equipe BD no ambiente de desenvolvimento, cada integrante das diversas equipes de cada módulo testava suas modificações para garantir que a base contemplava tudo o que foi requisitado. Ao encontrar erros, cada equipe fazia suas modificações e solicitava a modificação para a equipe do módulo BD produzir, novamente, uma nova versão do ambiente de desenvolvimento para que pudesse reiniciar o teste.

Ao final de todos os testes específicos de todos os módulos individualmente, era compilada uma versão da base de dados e dos executáveis do sistema, para que se executassem uma bateria de testes conhecida como “Testes Integrados”, onde se procuravam por erros supostamente criados a partir de mudanças realizadas em um determinado módulo, que impactariam na funcionalidade e desempenho de outros (o que acontecia com certa frequência). Ao se encontrar algum erro em decorrência destas interferências entre módulos, correções eram desenvolvidas e tudo era testado novamente.

Ao chegar na semana anterior à data da máster, era executado pela equipe de Gerência do Sistema o processo de preparação do servidor que iria assumir como produção (até então estava de reserva em *standby*) com a compilação da nova versão. Mais um teste era executado, desta vez chamado de “Teste de Pré-Produção”.

Na máster era executado um processo com diversos passos para que o sistema fosse migrado do servidor que estava em produção para o que estava em reserva, já configurado com a nova versão. Após este chaveamento, mais um teste era demandado, desta vez com o auxílio também dos operadores do centro de operação para a verificação se tudo estava em ordem, inclusive com as comunicações com os diversos pontos de contato do sistema, incluindo o ONS.

3.0 Embasamento Teórico e Boas Práticas

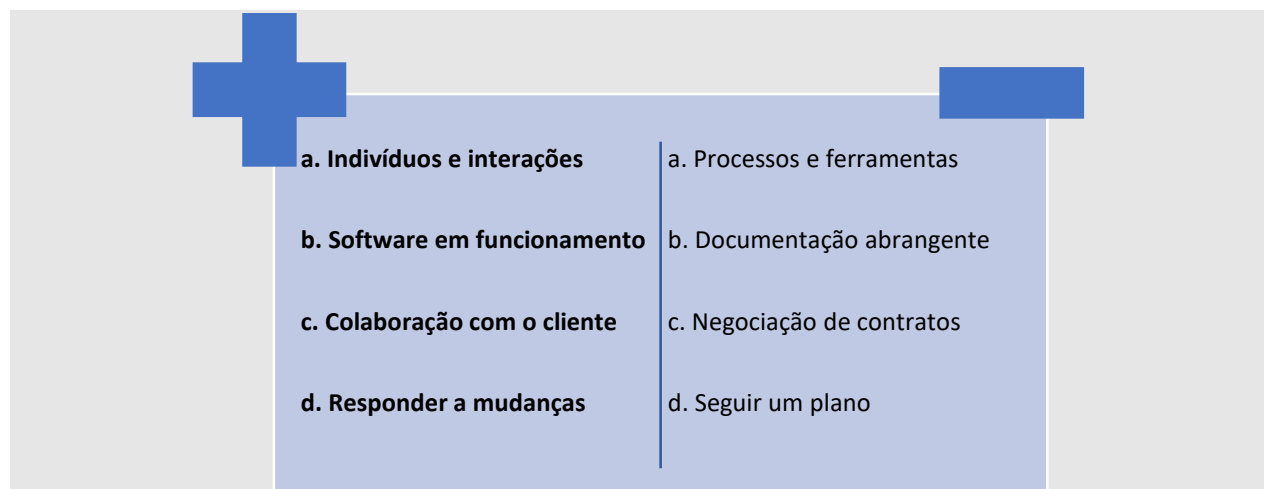
Comparando os processos de atualização de bases de dados com o de desenvolvimento de software, é possível notar que a manutenção de bases de dados se inicia a partir do escopo da obra ou da melhoria a ser implementada no campo, assim como o desenvolvimento de software inicia com um escopo de funcionalidades a serem implementadas. Depois é discutido como será desenvolvido este escopo e qual estratégia usar para cada requisito em ambos os processos.

É preciso se atentar ao cronograma e saber gerenciar bem os recursos para não atrasar a implantação. Testes são imprescindíveis para que se possa garantir que o que foi definido inicialmente tenha sido entregue e implementado corretamente, sem interferência em funcionalidades já testadas. Monitorar a qualidade do processo e do produto é crucial para que se possa estabelecer a melhoria contínua.

Para o processo de desenvolvimento de software, muitas equipes no mundo inteiro trabalham em soluções para estes problemas. Fazendo este comparativo com o tratamento de bases de dados, é possível trazer os avanços acadêmicos do processo de desenvolvimento de software para o de manutenção de bases de dados.

Nos últimos anos, as metodologias ágeis se consolidaram como um excelente modelo a seguir para a gestão do processo de desenvolvimento de software. Estas metodologias nasceram a partir do “Manifesto Ágil” (1) que tornou claras algumas das premissas básicas para nortear as estratégias de desenvolvimento de software.

Neste manifesto, os itens à direita de cada afirmação não são desprezados, são considerados importantes, porém os itens à esquerda são mais valorizados. Na realidade do processo de manutenção de bases de dados, é possível enxergar benefícios diretos em cada um dos valores descritos neste manifesto:



- a. A equipe trabalha melhor quando interage durante todo o fluxo do processo, diminuindo ruídos de comunicação e trabalhando junto para conseguir o objetivo.
- b. Neste caso “Software em funcionamento” poderia ser substituído por “Equipamentos integrados em operação remota” no caso do processo de manutenção de bases para sistemas EMS.
- c. No caso do processo de manutenção de bases, o cliente é o operador da sala de controle. Este cliente necessita de uma interface eficiente, todas as funcionalidades e automações em pleno funcionamento e comunicação constante das evoluções criadas. Formalidades são importantes visto a criticidade do processo, porém ter uma colaboração e feedback constante com os operadores é crucial para o sucesso deste modelo.
- d. Nos projetos de integração, mudanças de detalhes durante a execução não são incomuns e um processo burocrático pode trazer prejuízos para o resultado final da implantação do escopo.

A partir destas poucas (apenas 4) diretrizes, todas as metodologias ágeis se formam para guiar estratégias mais concretas de fluxo de desenvolvimento. Como base para o trabalho, foram adotados, principalmente, conceitos de 3 metodologias ágeis: *Lean Manufacturing* (2), *DevOps* (3) e *Kanban* (4).

Lean Manufacturing (ou “Manufatura Enxuta” como é traduzida em algumas bibliografias), é uma filosofia operacional que há décadas vem trazendo resultados expressivos. Não começou no desenvolvimento de software, mas sim na gestão industrial. O *Lean* visa tornar o processo de manufatura cada vez mais enxuto. É tratado como uma filosofia justamente por propor transformar a abordagem do processo inteiramente em relação a seu fluxo.

Em cada etapa do processo, o valor deve ser encarado pela ótica do cliente. Assim, é possível eliminar ou otimizar as etapas dos processos que pouco agregam qualidade ao produto ou serviço, deixando o fluxo mais eficiente. O fluxo é em volta do *valor*, que é uma palavra muito importante em *Lean*.

A origem do *Lean* é do famoso “Sistema Toyota de Produção”, criado no início da década de 1950 no Japão, na montadora japonesa, que utiliza seus preceitos até hoje. O objetivo do *Lean* é bastante claro: evitar perdas e desperdícios, eliminando da cadeia operacional as execuções que não agregam valor ao processo produtivo e ao produto final.

O objetivo ao analisar a cadeia é identificar o valor, mapear suas origens, criar um fluxo produtivo que potencialize essa entrega, e buscar a perfeição através da melhoria contínua. Um dos desperdícios mais criticado pelo *Lean* é o retrabalho. Se o processo propicia uma grande chance de erros acontecerem, o retrabalho é inevitável.

Com o *Lean* é possível, através de uma estrutura sistemática, eliminar o desperdício sem sacrificar a produtividade. São 8 classificações para os desperdícios do *Lean Manufacturing*:

- Transporte e logística: deslocamentos desnecessários de materiais, causando desperdício de tempo e recurso. Neste caso é possível traçar um paralelo com o processo de integração de ativos para o desperdício em etapas que dependem de insumos que não estão sincronizados com o cronograma.
- Superprodução: quantidades produzidas além do necessário. Este desperdício pode ser visto no processo de integração de ativos ao se trabalhar além do necessário para um determinado escopo, trazendo funcionalidades que são às vezes pouco utilizadas ou possíveis de serem substituídas por outras soluções nativas.
- Tempo de espera: materiais, equipamentos, informações e pessoas que ficam limitadas às etapas anteriores, que geram perda de tempo ou atraso nos processos. Este é o tipo de desperdício mais enxergado no processo de manutenção de bases de dados, pois em vários pontos do fluxo existem trabalhos que dependem de insumos que são saídas de outras etapas do fluxo. O “*handoff*” é um dos principais desperdícios no processo.
- Excesso de processamento: etapas ou processos que não são necessários e não agregam valor ao produto. Este desperdício é bem evidente na teoria, mas um dos mais difíceis de se tratar na prática. Com um processo que é balizado por muitos documentos (*compliance*) como o de manutenção de bases de dados EMS, é muito difícil se encontrar uma etapa que pode estar apenas “burocratizando” o fluxo, mas quando identificada e eliminada, o ganho de eficiência é evidente.
- Estoque e inventário: insumo acumulado em exagero, por erros de troca de informações internas ou problemas com entregas de fornecedores. Um fluxo de integração de ativos bem monitorado e com suas etapas bem definidas pode evitar este “congestionamento” de insumos em uma determinada etapa, trazendo o problema à tona o mais cedo possível para que se possa aumentar o esforço o quanto antes.
- Movimentação de Pessoal: Colaboradores que precisam se locomover sem necessidade, por demandas do processo produtivo. Isso pode acontecer também no processo de manutenção de bases. Digitalização de comunicações e clareza no processo evitam este tipo de desperdício.
- Defeitos: problemas que trazem prejuízos à indústria e aos clientes. Este desperdício é extremamente crítico para o processo de integração de ativos pois pode causar um dano muito grande para o processo de operação. Mesmo que se encontre o defeito nas etapas de desenvolvimento e homologação, existe um desperdício de tempo para a resolução deste problema. Automações e validações em etapas iniciais podem evitar este desperdício no processo de integração de ativos como um todo.
- Habilidades subutilizadas: aproveitamento superficial do potencial de cada colaborador. No processo de manutenção de bases de dados cada etapa do fluxo exige colaboradores com competências diferentes, saber encaixar o colaborador certo na etapa certa evita desperdiçar essas habilidades.

Uma das ferramentas utilizadas para a implantação e monitoramento dos fluxos traçados através das diretrizes do *Lean Manufacturing* é o *Kanban*. O *Kanban* é um método de gestão que permite maior visualização do fluxo de operação do processo, mostrando em qual etapa cada item em produção está.

O *Kanban* também surgiu no Sistema Toyota de Produção. Com foco no aumento da produtividade e na otimização dos processos, o Kanban busca conduzir cada tarefa por um fluxo predefinido de trabalho, indicando e limitando o

trabalho em andamento (WIP – *Work In Progress*). Em geral o conceito de *Kanban* pode ser definido pelos seguintes itens:

- O sistema visual: um processo definido em um quadro com colunas de separação, que permite dividir o trabalho em segmentos ou pelo seu status, fixando cada item em um cartão e colocando em uma coluna apropriada para indicar onde ele está em todo o fluxo de trabalho.
- Os cartões: que descrevem o trabalho real que transita por este processo
- A limitação do trabalho em andamento: que permite atribuir os limites de quantos itens podem estar em andamento em cada segmento ou estado do fluxo de trabalho.

O foco do *Kanban* é priorizar a produtividade e organização das entregas, propiciando um trabalho mais transparente e direcionado. Na prática, o Kanban traz um quadro visual dividido em colunas que definem basicamente o que está na entrada (a fazer), o que está sendo feito no momento (fazendo) e as tarefas que já foram executadas (feito). Um conceito importante do *Kanban* é o WIP (*Work In Progress*). Cada coluna possui um número WIP associado que mostra a capacidade do fluxo de trabalho do time naquele estágio, dando uma boa ideia geral de como está o processo comparando o número de cartões naquele seguimento com o seu WIP. Outra definição importante do *Kanban* é que as tarefas são “puxadas” (*pull*) para o estado de “fazendo”, criando um caminho claro de objetivo de levar as tarefas para o estado final (feito).

O *Kanban* traz uma maior transparência para o processo para todo o time e melhora a performance do dia a dia da equipe, melhorando a qualidade e a eficiência do produto final. Além de otimizar a produtividade, é possível priorizar as tarefas mais importantes, balanceando as demandas, reduzindo custos e incentivando a comunicação entre a equipe.

Outra metodologia ágil importante para o desenvolvimento de software que pode ser portada para o processo de manutenção de bases de dados de sistemas SCADA EMS é o *DevOps*. Esta metodologia consiste em integrar o time de Desenvolvimento (que codifica o software ou implementa a base de dados) com o time de Infraestrutura (“Operação” no conceito de TI).

O *DevOps* define as etapas do ciclo de desenvolvimento e operacionalização do produto (*software* ou sistema SCADA EMS) de modo a evoluir constantemente, com integrações rápidas e constantes, eliminando processos burocráticos e erros desnecessários.

A figura a seguir demonstra o ciclo de desenvolvimento:

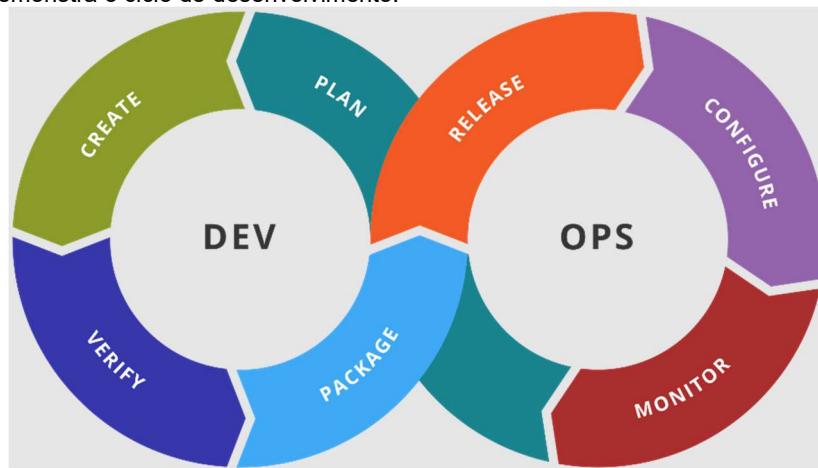


FIGURA 2: Ciclo *DevOps*

É possível ver que o processo de *DevOps* é cíclico e que o monitoramento do processo realimenta a fase de planejamento. As fases do ciclo *DevOps* podem ser definidas como:

- **Planejamento (*plan*):** Fase em que se planeja o que será implementado. No processo de edição de bases pode ser a fase em que o campo demanda uma alteração na base de dados do centro de operação por uma obra ou melhoria na instalação.
- **Criação (*create*):** Fase em que se desenvolve as alterações que foram planejadas. No processo de edição de bases é o momento em que se pega a base atual do sistema e incorpora as modificações propostas no planejamento.
- **Verificação (*verify*):** Fase de testes para homologar que o que foi pedido foi implementado. Nesta fase, ainda em processo de desenvolvimento, é feita a verificação que foram implementadas na base as modificações pedidas pelo planejamento. Pode ser feito um teste ponto-a-ponto (comissionamento com o

campo) nesta fase, ainda em ambiente de homologação, para a verificação de que tanto a base de dados do campo quanto a base de dados do sistema remoto estão condizentes.

- **Empacotamento** (*package*): Fase em que é gerada a imagem que define a versão. Nesta fase é feito o empacotamento das modificações que entrarão em homologação e produção.
- **Implantação** (*release*): Fase em que se unifica o produto para entrada em produção. No processo de manutenção de base, esta fase pode ser a que se configura o ambiente em *standby* para assumir a produção com a nova versão.
- **Configuração** (*configure*): Nesta fase se coloca o ambiente com a nova versão em produção. No processo de manutenção de base é o momento em que se chaveia o ambiente de produção com a versão antiga para o ambiente com a versão nova.
- **Monitoramento** (*monitor*): Nesta fase, já com o ambiente novo em produção, é feito o monitoramento dos parâmetros do sistema de modo a realimentar o processo de planejamento caso seja evidenciado alguma mudança a ser feita. No processo de manutenção de bases, pode ser o monitoramento de pontos que estejam com alguma falha, processos que se tornem menos eficientes, dentre outros parâmetros que possam alimentar o processo objetivando a melhoria contínua.

Um dos conceitos mais importantes para o processo de DevOps é o de CI (*Continuous Integration* – Integração Contínua) e o de CD (*Continuous Deploy/Delivery* – Implantação/Entrega Contínua) (5).

A Integração Contínua (CI) aborda os aspectos de desenvolvimento da base, integrando os diversos desenvolvedores do time de modo a não causar conflitos e erros no processo.

Já a Implantação/Entrega Contínua (CD) é o processo de se fazer com que as mudanças cheguem em produção o mais rápido possível, garantindo que o sistema esteja válido em todas as versões. É um processo extremamente dependente de automações para que seja eficiente. Automações estas que também passam por evoluções no ciclo de desenvolvimento provido pelo *DevOps*.

A combinação de CI/CD proporciona um ambiente mais claro de desenvolvimento em equipe, garantindo que os conflitos sejam resolvidos o mais rápido possível para que não impactem no processo e a entrega seja contínua.

Para a tarefa de se organizar as versões dos arquivos texto de entrada para a compilação da base do SAGE, foi utilizada a ferramenta **Git** que é um controle de versões distribuído. Com ele, todo o controle de versões é feito na máquina local de desenvolvimento de cada membro da equipe. Para enviar para o repositório central, é feito um processo de sincronização entre o repositório local e o remoto de modo a evitar e resolver possíveis conflitos.

Foi definida a utilização do **Gitlab** como ferramenta servidora do repositório central e que tem outras responsabilidades no processo como a auditoria de quem fez qual alteração e quando, com uma interface gráfica muito amigável e recursos de automação que facilitam a verificação do código e a compilação automática, ajudando na diretriz de integração contínua (CI) e na de entrega contínua (CD) através de processos automatizados executados por demanda para a etapa de entrega à produção.

Também é utilizado o Docker, que é uma aplicação OSS (*Open Source Software*) para a criação de ambientes virtuais padronizados, que facilitam a configuração de aplicações. Por trás do Docker está a tecnologia de *containers*, que permite empacotar a aplicação com tudo o que ela precisa para funcionar (*softwares*, bibliotecas, etc) sem o peso de outras camadas do sistema operacional que não são utilizadas pela aplicação, como em uma máquina virtual tradicional.

4.0 NOVO PROCESSO

No decurso de se implantar a metodologia *Lean* para o processo de manutenção de bases de dados SCADA EMS, é importante não se fazer uma mudança muito abrupta.

A primeira fase executada foi a de se mapear todas as etapas do processo antigo desde a concepção da obra ou melhoria até a implantação em produção no sistema SCADA EMS. É um estudo complexo, com interação com diversas equipes da empresa e com metodologias diferentes para cada time.

A partir deste levantamento inicial, foi feita uma análise crítica de qual etapa poderia ter um maior impacto no valor agregado na entrega do produto aos operadores da sala de controle. Foi escolhida a etapa de desenvolvimento da base pela equipe do IMSC e a aplicação de conceitos de *DevOps* para evolução desta fase, aproveitando o projeto de substituição do sistema SCADA EMS em curso.

4.1 TRATAMENTO DA BASE

A etapa de tratamento da base pelo IMSC foi a escolhida como a etapa inicial para se prover essa evolução no processo. Os conceitos de *DevOps* e de CI/CD são explorados para a elaboração deste processo como é explicado a seguir.

No novo processo, na fase de planejamento, são utilizados conceitos de *Kanban* para a determinação das tarefas a serem executadas em cada escopo de modificação da base. Estas tarefas passam por um fluxo na medida que vão sendo executadas pela equipe, desde o planejamento até a implantação em produção, liberando o escopo para a operação pelos operadores do centro de operação.

Na etapa de criação, a base de dados do sistema é modificada de acordo com a tarefa a ser tratada. Cada desenvolvedor manipula a base em sua console de desenvolvimento, de acordo com o que foi pedido na fase de planejamento.

Foi implantado um sistema de controle de versões Git para o gerenciamento dos arquivos texto das bases de dados SAGE, de modo a garantir que todos trabalhem com a versão correta da base. Ao se alterar a base localmente em suas consoles de desenvolvimento, a integração com a versão oficial do repositório Git da base é feita de maneira a evitar conflitos com outros membros da equipe. Esta ferramenta também possibilita a auditoria de modificações, apontando quando, quem e o que foi modificado em cada *commit* (versionamento).

A partir de imagem Docker SAGE, disponibilizada pelo Cepel, foi possível desenvolver uma ferramenta de validação da base local na própria console de desenvolvimento de cada integrante da equipe. Nestas máquinas (Windows) não existe o SAGE instalado, o que faz com que o uso deste *container* Docker facilite muito o trabalho. As mudanças locais de cada desenvolvedor podem ser validadas em alguns segundos, através de um script de compilação da base utilizando a imagem Docker, facilitando muito o processo de desenvolvimento.

Com a base já validada localmente, o desenvolvedor passa as modificações para o repositório central do Gitlab. Este processo nativamente rejeita qualquer conflito e garante que sejam resolvidos localmente antes de integrar com o repositório centralizado.

Para dificultar o aparecimento de conflitos na base de dados, também foi desenvolvida uma segregação dos pontos através de arquivos de *"include"* na base, dividindo por pastas cada ligação SCADA. Isso possibilita que cada integrante da equipe, ainda na fase de planejamento, divida as tarefas de modo a executarem escopos diferentes ao mesmo tempo, evitando a concorrência de edição de um mesmo arquivo.

Ao se integrar a base, o Gitlab dispara um teste para garantir que a base está íntegra, rejeitando a integração caso haja falha neste processo. Isso faz com que o repositório central esteja sempre íntegro e possíveis erros no desenvolvimento local não se propaguem a outros integrantes da equipe.

Todas estas etapas garantem o conceito de Integração Contínua (CI) de modo a garantir a consolidação de toda a base a cada modificação integrada no tronco principal do repositório do gitlab de maneira frequente.

Ao se obter o conjunto de modificações planejadas no repositório central, é possível executar o processo de implantação em produção da base. Um processo de implantação automatizado e validado garante que a entrega de modificações seja mais frequente e diminui em muito o represamento que pode afetar todo o fluxo.

Por demanda, é gerada uma *release* (compilação de versão) a partir da base corrente do repositório através de um processo automatizado. Esta é a etapa de Empacotamento e *Release* do *DevOps*. Com esta release feita, se implanta a versão no ambiente de homologação (QADS, ou *Quality Assurance Data System*) de modo a verificar em testes integrados manuais e automatizados as modificações para validação dos escopos tratados na versão.

Com a release validada, ela será implantada no ambiente de produção através de um processo automatizado que:

- Interrompe a base no ambiente até então em produção.
- Atualiza o ambiente de *standby* com a *release* já homologada
- Parte o processo de preservação dos dados do ambiente em produção para o ambiente de *standby*
- Chaveia o ambiente, tornando o ambiente até então em *standby* em produção e faz a ativação da base neste ambiente, liberando o sistema para operação.

Este processo garante que, se houver algum erro não encontrado no processo de desenvolvimento ou no de homologação, o *"rollback"* (voltar atrás com a versão antiga de produção) seja garantido através de um simples chaveamento de ambiente para o que está em *standby* (em produção antes do chaveamento).

Para a etapa de monitoramento do *DevOps*, observações levantadas pelos operadores, indicações de falhas em algum ponto do sistema ou em canais de distribuição, fazem com que seja realimentado o processo de planejamento para estudo e modificações objetivando a evolução e melhoria contínua.

O Estimador de Estados (EE) é um módulo de análise de redes elétricas que integra os sistemas EMS. Esta aplicação utiliza os dados dos diversos medidores do sistema, sua topologia corrente e parâmetros elétricos como impedância de linhas e transformadores para que se calcule um valor estimado para as diversas medidas elétricas em todos os pontos do SEP modelado.

O EE, ao calcular o estado corrente de todos os valores elétricos do SEP modelado, acusa quais medidas estão com estado suspeito (erros grosseiros). Estes podem ser resultantes de erros de modelagem topológica do sistema, erros de indicações de estado de equipamentos de manobra (aberto/fechado de disjuntores por exemplo) e/ou erros de indicação de medidas analógicas que podem ter falhas na modelagem de mudança de escala de transdutores ou multimedidores no caminho de aquisição do campo ao sistema SCADA. A monitoria destas indicações traz um caminho claro de realimentação do processo, indicando para a fase de planejamento a necessidade de estudo para a remodelagem de algum erro encontrado.

Além do EE, o ONS possui um relatório chamado Monitoramento da Qualidade e Disponibilidade das Informações (MQDI) cujo objetivo é auxiliar os agentes do SIN a avaliar e aprimorar a disponibilidade e a qualidade do conjunto de grandezas supervisionadas disponibilizadas para o ONS para a operação do SIN. O MQDI calcula e disponibiliza em um portal web indicadores de disponibilidade e qualidade dos pontos de supervisão de equipamentos do SIN.

Com posse do relatório do conjunto de informações com baixa qualidade no MQDI, o processo de planejamento pode ser munido com as informações necessárias para se estabelecer um estudo de modo a descobrir a origem do defeito que pode estar atrelado a problemas nos canais de comunicação com a remota do campo, falha na modelagem de aquisição ou falha na modelagem de distribuição.

5.0 COMPARAÇÕES

No processo de máster (antigo), havia um grande represamento de demandas pois as atividades despendiam muito esforço. Tinham interligações internas que fazia com que o processamento de algumas etapas tinha como entrada a saída de outras. Com isso o tempo de *lead* (período entre o surgimento de uma nova tarefa em seu fluxo de trabalho e sua partida final do sistema no *Kanban*) tinha média na casa de meses, ao invés de poucas semanas como no processo novo.

Além disso, para reforços e melhorias que tinham cronogramas finalizando no período entre duas datas de másters, o envio da base de dados com 60 dias de antecedência da máster anterior à energização (figura 1) trazia como consequência um retrabalho significativo, pois a base de dados enviada ainda não estava em um estágio “maduro” e, frequentemente, esta base do campo evoluía e tinha de ser corrigida na máster seguinte, o que poderia inclusive atrasar o início da operação do ativo pela companhia.

Para um período que se atravessa de grande valorização do processo de reforço e melhoria, este processo agia como uma “impedância” para a integração e entrada em operação dos ativos.

Todo este processo demandava uma equipe para cada módulo do fluxo, com especialidades cujos conhecimentos eram difíceis de se transmitir para novos membros, além de ser um sistema com um alto de grau de acoplamento entre os módulos e feito internamente pela equipe, sem um suporte da ferramenta pelos desenvolvedores (em grande parte já desligados da empresa).

Já no processo novo, a equipe se tornou multidisciplinar. Cada demanda pode ser puxada (*pull*) por qualquer membro da equipe na fase de planejamento para a execução. Caso a demanda seja de alguma tarefa com qual o membro da equipe não é familiar, outro membro da equipe é alocado em par para repassar o conhecimento. Isso promove nivelamento de conhecimento entre membros da equipe, que beneficia todos os estágios do fluxo. Se houver alguma quantidade de demanda além do esperado (acima do WIP) em alguma etapa do processo, a equipe pode absorver esta anormalidade, em geral, sem impactar em outros estágios do fluxo.

6.0 CONCLUSÃO

A identificação das semelhanças nas características e principalmente dos desafios dos processos de manutenção de bases de dados de sistemas SCADA EMS com os processos de desenvolvimento de software foi essencial para a motivação deste trabalho. Equipes em todo o mundo estão desenvolvendo melhores práticas para o desenvolvimento de software e algumas destas são possíveis de se adaptar para o processo de manutenção de bases de dados SCADA EMS e integração de ativos como um todo.

As automações e melhorias no processo trouxeram um nível muito menor de retrabalho, com um tempo de *lead* muito menor, seguindo a diretriz de entrega contínua.

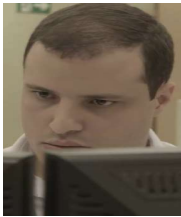
Para trabalhos futuros será continuada a análise crítica do fluxo do processo de integração de ativos como um todo, com direcionamento do *Lean Manufacturing*, avaliando desperdícios a serem eliminados e trabalhar em torno do aumento da eficiência do processo como um todo, etapa por etapa.

Este é apenas o primeiro passo para um processo mais ágil, mas que já se obtém resultados favoráveis e que podem ser implementados em outras companhias do setor elétrico.

BIBLIOGRAFIA

- (1) Manifesto para Desenvolvimento Ágil de Software – <https://agilemanifesto.org/iso/ptbr/manifesto.html> . Acesso em: 23/09/2021.
- (2) Lean Manufacturing: o que é, ferramentas e como implementar – <https://www.totvs.com/blog/gestao-industrial/lean-manufacturing/> . Acesso em: 23/09/2021.
- (3) Kanban: Conceito, como funciona, vantagens e implementação – <https://www.totvs.com/blog/negocios/kanban/> . Acesso em: 23/09/2021.
- (4) O que é DevOps? - <https://www.redhat.com/pt-br/topics/devops> . Acesso em 23/09/2021.
- (5) O que é CI/CD? - <https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd> . Acesso em 23/09/2021.

DADOS BIOGRÁFICOS



RAFAEL FRANCO CORDEIRO

Formado em Engenharia Elétrica com certificado de estudos em Computação pela UFMG (Universidade Federal de Minas Gerais), com especialização em sistemas elétricos de potência (CESEP - UFMG) e extensão em Engenharia de Software Moderna também pela UFMG. Trabalha no setor de Implantação e Manutenção de Sistemas de Controle (IMSC) do Centro de Operação do Sistema (COS) da Cemig GT e tem grande interesse nas áreas de manutenção de sistemas SCADA, engenharia de software, cultura ágil, inovação, automação e melhoria contínua de processos.

FABIO AUGUSTO DA SILVA ANTUNES

Engenheiro do Centro de Operação do Sistema da Cemig GT, Doutor em Engenharia Elétrica (2019) pela UNIFEI - Universidade Federal de Itajubá, universidade na qual também obteve os títulos de Mestre em Engenharia Elétrica (2012) e Especialista em Sistemas Elétricos (2009), graduado em Engenharia Elétrica pela UNESP - Universidade Estadual Paulista (2005), campus de Ilha Solteira. Atua na área de Operação de Sistemas Elétricos de Geração e Transmissão desde 2007, na Cemig GT.

CAROLINA CÂNDIDA DE OLIVEIRA

Engenheira eletricista pela PUC Minas (2009). Especialização em Docência na Educação Profissional e Tecnológica (2011), Senai RJ e Geoprocessamento (2015), PUC Minas. Possui MBA em Gestão de Negócios com ênfase em TI (2013), IETEC e extensão em Engenharia de Software Moderna (2020), UFMG. Quatorze anos de experiência nas áreas de operação, serviços de campo e pós-vendas na CEMIG Telecom. Atualmente é Engenheira do Centro de Operação Remota do Sistema na CEMIG GT, área de Integração e Manutenção do Sistema de Controle. cursando Transmissão e Distribuição de Energia Elétrica na PUC Minas e Ciência da Computação na Estácio.