

Lais da Silva Lira

***Validação De Software Para Laboratórios
O atendimento aos requisitos da qualidade do software (ISO/IEC 25010:2011)
não acolhe a completeza dos requisitos dos sistemas de gestão da qualidade
com relação ao uso de programas em laboratórios.***

Monografia apresentada ao Instituto de Química de
São Carlos da Universidade de São Paulo como
parte dos requisitos para a obtenção do título de
bacharel em química.

Orientador: Vitor Hugo Polisél Paces

São Carlos
2016

Agradecimentos

Primeiramente ao meu orientador, Vitor, por ter me dado esta oportunidade e por ter confiado em mim para trabalhar neste projeto, e que acima de tudo foi um grande amigo.

A minha família, que sempre me apoiou e me incentivou, estando comigo em todos os momentos da minha vida, inclusive quando eu quase desisti da graduação. Agradeço em especial o meu avô, que está presente apenas em espírito, nesta grande etapa da minha vida.

Ao meu namorado, Otávio, que esteve ao meu lado nos momentos bons e nos ruins, que sempre confiou em mim e me confortou com seus abraços e palavras de incentivo, e pelo seu amor incondicional.

Aos meus amigos, pelos momentos felizes e por todas as risadas e gordices compartilhadas ao longo da minha formação acadêmica.

E por fim, a que todos tanto direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

Resumo

A evolução do conceito de gestão de qualidade atrelada à evolução tecnológica lançou uma nova filosofia administrativa em um cenário mundial globalizado e exigente, o que tem sido fundamental para a conquista de novos consumidores. Face à esta situação, um problema surge: à medida que a empresa se expande, a quantidade de dados e informações também aumenta, o que dificulta a análise eficiente destes dados para tomada de decisão. Diante desse empecilho, frente ao custo decrescente da tecnologia, associado a sua maior facilidade de uso, a proposta é implementar um sistema de informação potencializado com a utilização da informática em todos os setores da empresa, inclusive nos laboratórios, sob forma de velocidade e confiabilidade. Para isso, os softwares que tratam esses dados e informações devem ser testados e validados antes da entrega final, ou seja devem estar funcionando de acordo com seu uso previsto, caracterizando um software. Entretanto, ter um software de laboratório validado de acordo com a norma de qualidade de produto de software (ISO/IEC 25010:2011), que garante um produto final de qualidade, não necessariamente garante a qualidade dos requisitos de gestão do laboratório.

No presente trabalho foi criado um roteiro de apoio à validação de software para laboratórios, auxiliando o desenvolvedor do programa e um guia auxiliando o auditor, baseado na norma ISO/IEC 17025:2005. Sendo um complemento à norma ISO/IEC 25010:2011, já que ela sozinha não satisfaz os requisitos de gestão de qualidade em laboratórios, por ser uma norma muito genérica.

Abstract

The evolution of the concept of quality management linked to technological development has launched a new administrative philosophy in a globalized world, which has been fundamental on the conquest of new consumers. In the face of this situation, a problem arises: when the company grows, the amount of data and information also increase, which hinders the effective analysis of these data for decision-making. Before this setback, with the declining cost of the technology, coupled with their ease of use, the proposal is to implement an information system powered with the use of computerization in all sectors of the company, including in the laboratories, as mean of speed and reliability. To do this, the software that handle all data and information shall be tested and validated before the final delivery, the software must be working according to its intended use. However, having a lab software which was validated according to the product quality standard (ISO/IEC 25010:2011), which guarantees a quality product, does not necessarily is in accordance with the quality management requirements of the laboratory.

In the present work, it was created a laboratory software validation support screenplay, helping the software developer and a guide which assist the auditor, based on ISO/IEC 17025:2005. The screenplay is a supplement to ISO/IEC 25010:2011, since it alone does not satisfy the requirements of quality management in laboratories, because it is a very general standard.

Sumário

Introdução.....	6
Ojetivos.....	19
Roteiro de apoio à validação de softwares para laboratoriais para desenvolvedores.....	20
1. Termos e definições	20
2. Requisitos Gerais	21
2.1. Procedimentos Operacionais Padrão (POP) para validação de softwares	21
2.2. Plano de validação	21
3. Controle de usuário e acesso.....	23
4. Testes de software	24
4.1. Generalidades	24
4.2. Objetos ou funções que realizam cálculos.....	28
4.3. Objetos relacionados a banco de dados	28
4.4. Objetos que fazem uso de condicionais.....	30
5. Testes de planilhas eletrônicas	31
5.1. Generalidades	31
5.2. Para funções que realizam cálculos.....	32
5.3. Para funções condicionais	32
6. Manutenção/Backup.....	33
7. Relatório de validação e demais registros	34
Tutorial técnico de avaliação da validação de softwares para avaliadores.....	35
1. Generalidades	35
2. Requisitos gerais	36
2.1. Procedimentos Operacionais Padrão para validação de softwares	36
2.2. Plano de validação	36
3. Controle de usuário e acesso.....	36
4. Testes de softwares	37
4.1. Generalidades	37
4.2. Objetos ou funções que realizam cálculos.....	37
4.3. Objetos relacionados a banco de dados	37
4.4. Objetos que fazem o uso de condicionais.....	39
5. Testes de planilhas eletrônicas	39
5.1. Generalidades	39
5.2. Para funções que realizam cálculos:.....	39
5.3. Para funções condicionais:	40
6. Manutenção/Backup.....	40
7. Relatórios e demais registros	41
8. Constatações.....	41
Conclusões.....	42
Referências bibliográficas	43

Introdução

Um breve contexto histórico

Ao iniciarmos uma breve introdução histórica do conceito de “qualidade”, podemos nos remeter às primeiras relações comerciais estabelecidas entre o homem, criando a relação de cliente e fornecedor, apesar de não existir uma gestão de qualidade propriamente dita nesta época, o conceito de qualidade sempre esteve implícito nestas relações comerciais, uma vez que o homem sabia que para vender seus produtos e serviços seria necessário atender as necessidades dos clientes.^[1]

Com a Revolução Industrial, a partir do século XVIII, a mão de obra dos trabalhos manuais foi substituída pelo trabalho mecânico, tornando assim, necessária a inspeção de todos os produtos, o que deu origem ao modelo de Taylorismo, no século XIX, criada pelo engenheiro mecânico e economista norte americano, Frederick W. Taylor. A principal característica desse sistema é a organização e a divisão de tarefas em uma empresa, com o objetivo de obter o máximo de rendimento e eficiência com o mínimo de tempo e atividade.

No fim da Segunda Guerra Mundial (1939 – 1945), ocorreu um grande salto na evolução do conceito de gestão de qualidade. O Japão derrotado e totalmente devastado pela guerra, e com uma dívida a ser paga começou a investir em suas indústrias.

Como os japoneses dependiam das exportações para conseguir comprar praticamente tudo que necessitavam e se viram diante do desafio de vender a outros mercados com preços menores, produtos de qualidade igual ou superior, os japoneses aplicaram conceitos de Deming e Juran, que ao invés de encontrar e eliminar peças defeituosas buscava evitar que os defeitos ocorressem.^[1]

A partir da aplicação dos conhecimentos administrativos de Deming e Juran, o Japão com dificuldades econômicas e pobres de recursos naturais, devastado pela guerra, se reergueu e se reestruturou economicamente a partir do aumento da produtividade, tecnologia e qualidade de seus produtos.

O próximo grande passo da história da qualidade pode ser chamado de “normalização”, com a criação da ISO 9000, o que houve foi nem tanto uma mudança de conceitos ou abordagem (embora tenha ocorrido), mas uma popularização impressionante em meio às indústrias das certificações dos “sistemas de garantia da qualidade” segundo os padrões adotados internacionalmente.

Gestão de qualidade nos dias atuais

Dentro do contexto mundial, a busca da competitividade tem exercido um papel relevante para o aprimoramento e conquista de novos mercados. A qualidade está sendo visualizada como uma forma de gerenciamento que, quando implementada, visa melhorar de modo contínuo o desempenho organizacional.

Em busca de sempre garantir a qualidade dos produtos e serviços, a ISO (*International Organization for Standardization*) passou a ser a organização mais atuante no cenário mundial, sendo a série ISO 9000, a mais representativa.

A ISO é uma organização internacional independente não governamental composta de 162 países membros, publicou mais de 20.500 padrões internacionais, abrangendo quase todos os aspectos de tecnologia e indústria.^[2]

Estas normas podem certificar produtos e serviços em várias organizações do mundo, realizando uma padronização baseada em documentos que fornecem diretrizes de como fazê-la.

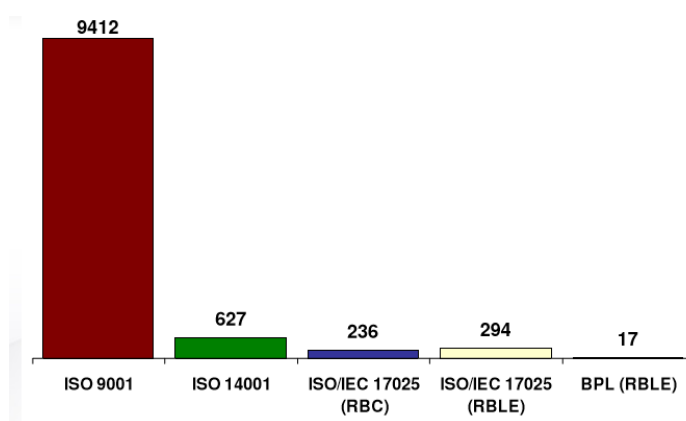
Podemos destacar alguns benefícios obtidos com o processo de certificação ^[3], para este presente trabalho, usaremos a palavra “certificação” de uma forma mais ampla, abrangendo os conceitos de acreditação (para laboratórios que atendam a ISO/IEC 17025) e reconhecimento (no caso de laboratórios que pertençam ao programa de reconhecimento da conformidade aos princípios das Boas Práticas de Laboratório – BPL):

- ✓ Grande oportunidade para impulsionar a imagem da organização;
- ✓ Mudança de foco da correção para a prevenção;
- ✓ Redução de desperdícios e custos.

É inegável que a certificação é uma ferramenta fundamental para as organizações que querem conseguir destaque no cenário mundial. Afinal, a organização que procura se adequar a alguma norma e, por consequência dessa adaptação, obtém e mantém a certificação, cria um grande diferencial competitivo frente aos seus concorrentes.

A Figura 1 é um reflexo da globalização. Com a abertura de novas fronteiras nacionais para o mercado internacional, o intercâmbio de produtos e serviços tem se intensificado, mostrando a aceitação dos acordos assinados (normas ISO). Desta maneira destaca-se a expressão: “testada uma vez, aceita em qualquer lugar”. [4]

Figura 1 - Empresas certificadas e laboratórios acreditados no Brasil por diferentes sistemas de gestão, cadastradas no INMETRO (2009) ^[1]



Fonte: Olivares, I.R.B., 2009

Gestão de qualidade nos laboratórios

Como foi possível observar a gestão de qualidade se encontra em todos os processos industriais de uma corporação, uma vez que o mercado está se tornando cada vez mais competitivo. Da mesma forma, os laboratórios também sofrem essa pressão para garantir a qualidade dos seus ensaios e calibrações.

Os laboratórios devem produzir resultados confiáveis para serem usados por seus clientes para o propósito planejado. [5]

O laboratório deve estabelecer implantar e manter um sistema de gestão de qualidade apropriado ao escopo de suas atividades, então a utilização de um sistema de gestão formalmente avaliado como, por exemplo, acreditação na

ISO/IEC 17025 ou as boas práticas de laboratório (BPL), podem auxiliar na padronização de todo o sistema de gestão de qualidade dentro dos laboratórios e, assegurando resultados de ensaio e/ou calibração confiáveis.^[6]

Além disso, como esses certificados são reconhecidos pelos organismos de acreditação dos países vinculados a organismos internacionais, que promovem acordo de reconhecimento mútuo, como o ILAC (International Laboratory Accreditation Cooperation), por exemplo, isso torna ainda mais fácil a comercialização de produtos e serviços prestados e também de informações.

Como pode ser bem observado, o conceito de qualidade evoluiu desde a simples adequação ao padrão para a adequação às necessidades do cliente.^[7]

Os principais fatores da implantação de um sistema de gestão de qualidade nos laboratórios e acreditá-los são listados a seguir ^[1]:

- ✓ Garantia de resultados confiáveis;
- ✓ Assegurar profissionais competentes;
- ✓ Assegurar instalações adequadas;
- ✓ Prestação de serviços padronizados.

A evolução da gestão de qualidade lançou uma nova filosofia administrativa e uma importante vantagem competitiva no mundo globalizado. Com a evolução tecnológica foi natural a absorção da automação dos principais processos utilizados.

A medida que a organização expande, a quantidade de dados e informações consequentemente aumenta; porém, como esses dados geralmente ficam dispersos, torna-se difícil extraí-los e analisa-los de forma eficiente.

Face esta situação, às novas tecnologias, às novas formas de organização e os novos conceitos de administração, novas propostas foram feitas para remediar esse problema, como por exemplo: sistemas de informação.^[7]

Importância do sistema de informação no sistema de gestão de qualidade

Sistema de Informação, na essência, não é específico à área de informática, apesar do termo sugerir isso. Esses sistemas referem-se aos processos existentes dentro da organização, empresa ou laboratório, que constituem seu funcionamento e levam esse nome, porque o resultado deles é exatamente isso: "informação". Como exemplo, podemos citar o sistema de controle de estoque, que traz como resultado uma série de relatórios, direcionados a etapas específicas dentro do processo funcional da empresa, servindo como base para a tomada de decisões. São, portanto, simplesmente a maneira de coletar, organizar e distribuir informações pela empresa.^[8]

Em um sistema de gestão da qualidade é importante que os procedimentos, instruções, especificações entre outros documentos normativos sejam documentados e os resultados dos processos organizacionais sejam registrados. Assim, observa-se a importância da gestão de informações para a gestão de qualidade.^[9]

Sistemas de informação são de extrema importância na área administrativa, e muitas vezes, esses sistemas podem ser potencializados com a utilização da informática, sob a forma de velocidade e confiabilidade no processo de gerar informação. A informática é uma ferramenta, utilizada para auxiliar estas atividades da empresa.^[8]

Atividades e trabalhos que antes eram feitos em dias, semanas ou até meses, podem ser informatizados e serem realizados em horas ou minutos, economizando tempo e, conseqüentemente, dinheiro com custos operacionais.

O fluxo de informações é um elemento de grande importância nas indústrias e nos laboratórios. Pedidos de clientes, necessidades de insumos, movimentação no estoque, documentação de transporte e faturas são algumas das formas mais comuns de informações.

Normalmente, o fluxo de informações baseia-se em papel, resultando em uma transferência de informações lenta, pouco confiável e propensa a erros. O custo decrescente da tecnologia, associado a sua maior facilidade de uso, permitem aos

gestores empresariais poder contar com meios para coletar, armazenar, transferir e processar dados com maior eficiência, eficácia e rapidez.^[10]

Dentre as vantagens da informatização, podemos citar ^[11]:

- ✓ Melhora as informações para tomada de decisão;
- ✓ Automatiza tarefas rotineiras;
- ✓ Aumenta a capacidade de reconhecer erros mais cedo;
- ✓ Aumenta a produtividade e competitividade.

Outro ponto importante para um processo de informatização bem-sucedido é a manutenção dos softwares que irão tratar as informações, pois alguns problemas no software não são descobertos antes que ele esteja sendo realmente utilizado e estes problemas devem ser evitados.^[12]

Qualidade de software

Antes de definir a importância da qualidade do software, é necessário entender o que é software:

Software é uma sequência de instruções escritas (componentes lógicos) que são interpretadas pelo computador com o objetivo de executar uma tarefa específica. De forma mais simples, são programas que permitem realizar determinadas atividades em um computador. Por exemplo, programas como o Word, Excel, PowerPoint, os navegadores, os jogos, entre outros.^[13]

No desenvolvimento de software, todos os defeitos encontrados são humanos e, apesar do uso dos melhores métodos de desenvolvimento, ferramentas ou profissionais, permanecem presentes nos produtos, o que torna a atividade de teste muito importante durante o desenvolvimento de um software.^[14]

Muitas são as causas-raiz destes defeitos: ^[14]

- ✓ A especificação pode estar errada ou incompleta;
- ✓ A especificação pode conter requisitos impossíveis de serem implantados devido a limitações de hardware ou software;
- ✓ A base de dados pode estar organizada de forma que não seja permitido distinguir os tipos de usuário;

- ✓ Pode ser que haja um defeito no algoritmo de controle dos usuários.

Estes problemas encontrados na construção do software podem acarretar em outros problemas de utilização do software, que são listados a seguir: ^[15]

- ✓ Programas que são descartados;
- ✓ Programas que simplesmente param de funcionar;
- ✓ Programas que não fazem exatamente o que seria esperado;

Levando em conta as variáveis: produtividade, tempo e custo de desenvolvimento, iniciativas devem ser tomadas no sentido de encontrar e corrigir defeitos tão logo sejam introduzidos. Uma solução que tem se mostrado eficiente e de baixo custo é a revisão dos artefatos produzidos ao longo do processo de desenvolvimento de software.

Antes de oferecer o produto para o cliente, a organização deve validar a operação do mesmo, de acordo com seu uso previsto e especificado no contrato, atendendo os requisitos do cliente. No curso desta validação, auditorias de avaliação podem ser realizadas, quando necessário. ^[14]

Importância da validação de softwares

Como foi descrito anteriormente, o objetivo da verificação e validação de softwares (V&V) é assegurar que o programa cumpra com suas especificações, atendendo as exigências do cliente.

A verificação consiste em uma atividade que envolve a análise de um sistema certificando se este atende aos requisitos funcionais e não funcionais, ou seja, visa assegurar que o software seja desenvolvido de maneira apropriada: ^[16]

“Estamos construindo corretamente o produto?”

Enquanto a validação é a certificação de que o sistema atende às necessidades e expectativas do cliente, ou seja, visa assegurar que o software corresponda aos requisitos estabelecidos: ^[16]

“Estamos construindo o produto certo?”

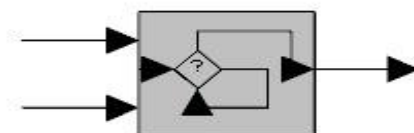
Como se pode observar, o processo de validação e verificação, não são processos separados e independentes. Durante este processo, podem ser utilizadas outras técnicas em conjunto, como a inspeção e/ou o teste de softwares.

A inspeção de softwares é uma verificação estática, isto é, o ato de analisar as representações do sistema: documentos de requisitos, diagramas e código-fonte. É recomendável que a inspeção seja feita em todas as etapas do ciclo de vida dos softwares. Por outro lado, o teste de software é uma verificação dinâmica, apenas é possível testar o programa quando o protótipo ou uma versão executável esteja disponível.

Os testes de software são processos importantes para a conclusão do produto, com o objetivo de verificar se existem falhas. Se existirem, identifica-las e corrija-las antes da entrega final. As técnicas mais utilizadas para se testar softwares são: técnica funcional e estrutural.^[17]

- ✓ **Técnica Estrutural (teste caixa-branca):** Esta técnica avalia o comportamento interno dos componentes do software (Figura 2), isto é, ela trabalha diretamente sobre o código-fonte.

Figura 2 - Técnica de teste estrutural



Fonte: Revista Engenharia de Software: Qualidade de software ^[18]

- ✓ **Técnica Funcional (teste caixa-preta):** Nesta técnica, os componentes de software são testados como se fosse uma caixa preta, isto é, o comportamento interno do programa não é considerado (Figura 3). Neste tipo de técnica, dados de entrada são fornecidos, o teste é executado e o resultado (dados de saída) é obtido e comparado a um resultado esperado previamente conhecido.

Figura 3 - Técnica de teste funcional



Fonte: Revista Engenharia de Software: Qualidade de software ^[18]

O processo de verificação e validação, incluindo os testes e inspeções de softwares acarretam em muitos benefícios para a organização. Desta forma observa-se a importância de se investir na validação de softwares, pois os testes a serem realizados reduzem o número de erros e defeitos, acarretando num processo de qualidade.

Além disso, quanto mais cedo o problema for encontrado, menor será o custo da correção, segundo a regra 10 de Myers, que estabelece que o custo da correção de defeitos sobe exponencialmente conforme a fase em que o defeito é detectado, ou seja, um defeito encontrado em produção custa muito mais do que se fosse encontrado na fase de análise ou em modelos de dados.^[13; 19]

Qualidade de software x Qualidade em laboratórios

Como mencionado anteriormente, a validação de software é de extrema importância para o laboratório, pois tem potencial de reduzir os problemas provenientes do desenvolvimento do software (sejam problemas de lógica ou no código-fonte). Sendo assim, é capaz de minimizar a ocorrência de erros no produto final ou no serviço em que é empregado, acarretando a diminuição de custos.

O uso de softwares em laboratórios que são ou pretendem ser acreditados ou reconhecidos deve seguir respectivamente a ISO/IEC 17025 ou a BPL.

Considerando inicialmente a ISO/IEC 17025, com relação ao requisito 5.4.7 (Controle de dados) determina que os softwares devem ser testados e validados. Estes testes e validações devem atender especificamente algumas regras destacadas nos subitens específicos.^[20]

Assim, o requisito 5.4.7.2 menciona que quando ocorre a aquisição, o processamento, o registro, relato, armazenamento ou recuperação de dados de ensaio ou calibração, o laboratório deve assegurar que o software desenvolvido pelo usuário esteja documentado em detalhes suficientes e apropriadamente validado, como adequado ao uso.^[20]

Neste mesmo item é mencionada a necessidade de se estabelecer e implementar procedimentos para a proteção dos dados; incluindo no mínimo a integridade e confidencialidade da entrada ou coleta, armazenamento, transmissão dos dados.^[20]

A norma também considera que os softwares comerciais de prateleira (processadores de texto, banco de dados, programas de estatística) usados pelo laboratório são considerados como validados, porém, convém que as configurações e modificações feitas nestes softwares sejam validadas segundo 5.4.7.2. Assim, consideremos, por exemplo, uma planilha desenvolvida em Microsoft ExcelTM. O software em si (Excel) é considerado validado, porém o usuário ao utilizá-lo como ferramenta para construir uma planilha poderá incluir fórmulas, cálculos e dados que podem conter erros. Ou seja, a planilha construída não é considerada validada e, portanto, necessita de verificações e testes que garantam a confiança de seu uso.

Outros requisitos da ISO/IEC 17025 que descrevem a respeito do uso de sistemas informatizados, mostrando a importância da rastreabilidade dos dados independentemente da mídia em que se encontram, sejam papeis ou mídias eletrônicos. Estes requisitos estão descritos nos itens 4.3 (Controle de documentos) e o 4.13 (Controle de registros).

Sobre o item 'Controle de documentos', a norma descreve que "o laboratório deve estabelecer e manter procedimentos para controlar todos os documentos que fazem parte do seu sistema de gestão tais como software" e o requisito 4.3.3.4 descreve que "devem ser estabelecidos procedimentos para descrever como são realizadas e controladas as alterações nos documentos mantidos em sistemas computadorizados".

Para o requisito 'Controle de registros' é mencionado que (requisito 4.13.1.1) "o laboratório deve estabelecer e manter procedimentos para identificar, acessar,

arquivar, armazenar, manter e dispor os registros técnicos e de qualidade” e que estes podem estar em quaisquer meios (papel ou eletrônico).

O requisito 4.13.1.4 relata “o laboratório deve ter procedimentos para proteger e fazer cópias de segurança dos registros armazenados eletronicamente e prevenir o acesso ou emendas não autorizados nesses registros”.

A respeito da NIT-DICLA-035 (BPL), que regula os Princípios de Boas Práticas em Laboratório, no seu requisito 8 (Plano do estudo) também cita a respeito da rastreabilidade dos dados eletrônicos: “Dados gerados através de entradas diretas de um computador devem ser identificados no dia da entrada dos dados pelas pessoas responsáveis por estas entradas. Sistemas computadorizados devem sempre prover rastreabilidade possibilitando uma auditoria completa para mostrar todas as alterações dos dados sem mascarar os dados originais. Deve ser possível associar todas as mudanças dos dados com as pessoas que fizeram as alterações, por exemplo, usando assinaturas com data (eletrônica).”.^[21]

Além disso, os guias apresentados baseiam-se em muitos requisitos exigidos pela NIT DICLA 038 (aplicação dos princípios da BPL aos sistemas informatizados), que diz que: “todos os sistemas informatizados que são utilizados para produzir, medir, ou avaliar dados com fins regulatórios devem ser desenvolvidos, validados, operados e mantidos de acordo com os princípios da BPL”.^[22]

Alguns dos requisitos que são exigidos pela NIT DICLA 038 dizem respeito à manutenção (“quando estas atividades de manutenção causarem a modificação do software poderá ser necessário validar de novo o sistema”); rastreabilidade dos dados (“deverá se possível associar a cada modificação à pessoa que realizou a alteração por meio de assinaturas eletrônicas”); segurança (incluindo segurança física, lógica e backup); validação dos sistemas informatizados, que diz que os sistemas devem estar aptos às tarefas para quais estão destinadas. A presente norma ainda descreve que parte da documentação relativa ao uso dos sistemas informatizados, deve estar na forma de procedimentos operacionais padrão.^[22]

Existem similaridades de requisitos quando comparados ambos os sistemas de gestão de qualidade para laboratórios. Estes demonstram, portanto, a

importância de se possuir todos os softwares presentes no laboratório devidamente validados.

A norma ISO/IEC 25010:2011 é uma norma para qualidade de produto de software, a qual define um conjunto de atributos divididos em oito características fundamentais com o objetivo de padronizar a avaliação desta qualidade de software.

Segundo a ISO/IEC 25010:2011, as oito características de qualidade que todos os softwares devem possuir, são listadas a seguir: ^[23]

- ✓ Adequação funcional
- ✓ Eficiência de desempenho
- ✓ Compatibilidade
- ✓ Usabilidade
- ✓ Confiabilidade
- ✓ Segurança
- ✓ Manutenção
- ✓ Portabilidade

De acordo com a NBR ISO 9000:2015, qualidade é o grau no qual o conjunto de características inerentes satisfaz os requisitos,^[24] ou seja, pode-se afirmar que se algum produto ou serviço atende aos requisitos especificados pela norma este mesmo produto ou serviço possui a qualidade desejada. Portanto, para se obter um software de qualidade deve-se cumprir o que a ISO/IEC 25010:2011 propõe, ou seja, o software deve satisfazer as características desejáveis quando usado em condições específicas.

Clientes exigem produtos com características que satisfaçam suas necessidades e expectativas, o cliente sempre determinará a aceitabilidade do produto. ^[24]

A qualidade pode, então, ser medida através do grau de satisfação em que se avaliam determinado produto ou serviço. No entanto, esse produto ou serviço pode ter qualidade e serem apropriados para alguns clientes e para outros nem tanto.

Conceituar dessa forma o termo “qualidade de software” no caso do presente trabalho se torna uma tarefa complicada, pois pode-se afirmar que um software submetido aos processos de verificação/validação de acordo com a norma ISO/IEC 25010 não necessariamente garante a qualidade dos requisitos de gestão de qualidade no laboratório. Torna-se necessário o atendimento dos requisitos específicos mencionados anteriormente e presentes nas normas de gestão da qualidade para laboratórios. Isto porque, a qualidade do software não está vinculada ao atendimento dos requisitos dos sistemas de gestão de qualidade para laboratórios.

Dessa forma, este trabalho apresenta um roteiro de apoio à validação de softwares laboratoriais para desenvolvedores e um tutorial para auditores e avaliadores de sistemas de gestão da qualidade em laboratórios no que diz respeito ao uso de softwares e registros eletrônicos. Enquanto o primeiro deve ser utilizado durante o desenvolvimento do software e sua pré-utilização, o segundo é somente um adicional para nortear o avaliador, pois normalmente o mesmo não tem profundos conhecimentos de informática e pode não saber como utilizar somente o primeiro.

Portanto o presente trabalho foi realizado considerando que apenas as normas de qualidade de softwares não são suficientes para validar um software para laboratório com um sistema de gestão implantado, sendo assim, este roteiro é complementar à ISO/IEC 25010 no que diz respeito à laboratórios.

Objetivos

O objetivo do presente trabalho é a criação de um roteiro de apoio à validação de softwares, auxiliando o desenvolvedor (programador) ao atendimento dos requisitos de gestão da qualidade para laboratórios. Contempla ainda um tutorial técnico de apoio para auxiliar o avaliador (auditor) na tarefa de obter evidências objetivas de que o software em questão é validado. Este trabalho baseia-se no atendimento a norma ISO/IEC 17025 e a BPL (NIT DICLA 035 e as demais normas de apoio).

ROTEIRO DE APOIO À VALIDAÇÃO DE SOFTWARES LABORATORIAIS PARA DESENVOLVEDORES

1. Termos e definições

Software desenvolvido: Um programa que possui uma sequência lógica interpretada pelo computador que tem como objetivo executar determinada tarefa fazendo uso de determinada linguagem de programação.

Plano de validação: Estabelece uma linha de análise do software, na qual determina-se os objetos/funções passíveis de exame e os respectivos testes a serem aplicados para a validação do software desenvolvido ou adquirido. É específico para cada software em questão.

Controle de versões: O controle de versão gerencia todos os itens (módulos, subprogramas, funções...) que compõe o software disponibilizando uma identificação única a qual controla o conjunto de alterações realizadas.

Relatório de validação: Registro que contém as conclusões da validação do software após o emprego do plano de validação. É composto por evidências da realização dos testes realizados, o resultado destes e a conclusão respectiva a possibilidade de utilização do software em questão.

Código fonte: Conjunto de palavras e símbolos dispostos de maneira lógica que formam as funções do software, fazendo com que este desempenhe sua tarefa.

Células: Encontro de linhas e colunas em uma determinada tabela de uma planilha eletrônica. Cada célula apresenta um endereço (referência) único.

Teste funcional: Tipo de teste que dados de entrada são fornecidos, o programa é executado e o resultado é obtido e comparado a um resultado esperado previamente conhecido.

Macro: Sequência de comandos que podem ser utilizados sempre que o usuário precisar executar uma determinada tarefa, ou seja, um atalho para determinadas instruções.

2. Requisitos Gerais

O software somente estará apto a ser utilizado após: a conclusão da validação, emissão do relatório de validação e do respectivo procedimento de uso.

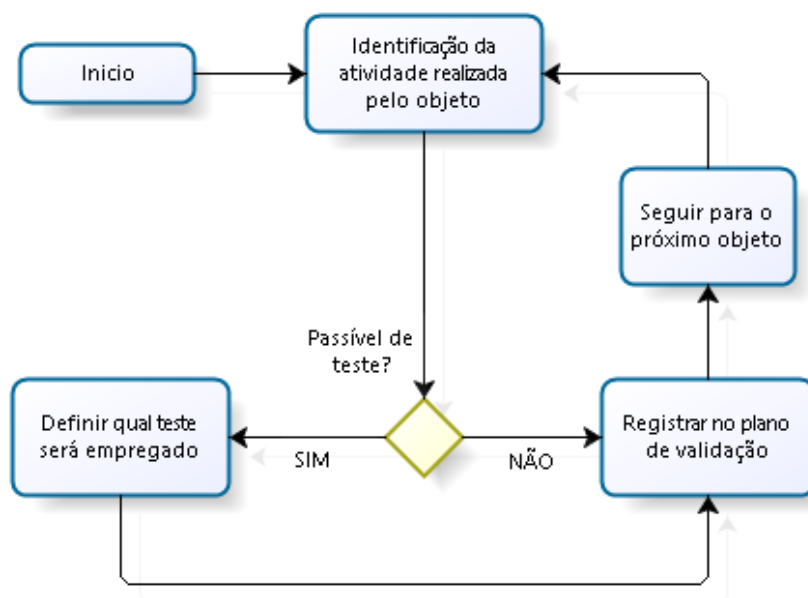
2.1. Procedimentos Operacionais Padrão (POP) para validação de softwares

- 2.1.1. Devem existir política e procedimentos definidos que determinam como executar a validação de softwares desenvolvidos para laboratórios ou adquiridos por terceiros. Todos os softwares devem possuir procedimentos para seu uso.
- 2.1.2. Para softwares adquiridos por terceiros, onde o laboratório não tenha acesso ao código fonte, devem ser realizados testes funcionais devidamente procedimentados que comprovem a adequação do mesmo ao uso do laboratório de acordo com o item 4 (Testes de software).
- 2.1.3. Quando aplicável, deve existir procedimentos que descrevam o processo de desenvolvimento do software e de controle das versões do mesmo.
- 2.1.4. Deve-se executar a validação de acordo com o procedimento operacional padrão e o plano de validação (descrito no item 2.2) relatando os resultados em registros apropriados.

2.2. Plano de validação

- 2.2.1. Com o objetivo de confirmar que o software esteja funcionando corretamente e que atende o uso pretendido, o laboratório deve elaborar um plano de validação para cada software desenvolvido ou adquirido de terceiro.
- 2.2.2. O plano de validação deve descrever os testes para todos os objetos e funções passíveis de avaliação relacionadas à validação de softwares desenvolvidos ou adquiridos por terceiros. Deverão ser avaliados os objetos e funções do software quanto à necessidade de testes, os quais irão compor o plano de validação, de acordo com o fluxograma apresentado na Figura 4.

Figura 4 - Fluxograma para análise de objeto ou função para verificar se o mesmo deve ser testado e incluído no plano de validação



Fonte: Autoria própria

2.2.3. No plano de validação deve constar, no mínimo:

- a) caracterização do software contemplando sua finalidade;
- b) a linguagem de programação utilizada;
- c) controle de desenvolvimento, alteração e implementação de novas versões;
- d) requisitos mínimos de sistema (incluindo hardware, software, bibliotecas, complementos e a definição de plataforma ou sistema operacional);
- e) todos os testes a serem realizados de acordo com as funções de cada objeto e respectivo método de aplicação, de acordo com o descrito no item 4 (Testes de software).

Nota: para softwares adquiridos de terceiro alguns itens mencionados podem não ser aplicáveis.

2.2.4. Todas as modificações realizadas no software após a validação tornam necessárias novas validações (parciais ou totais, de acordo com o impacto destas modificações) e devem seguir os presentes requisitos.

- 2.2.5. Quando forem implantadas novas versões do software, o novo plano e o relatório de validação dos novos testes realizados devem ser mantidos juntamente com os documentos e registros emitidos quando da validação das versões anteriores.
- 2.2.6. Os testes a serem realizados devem ser definidos por janelas, módulos, funções ou objetos, dependendo de cada caso, descritos no item 4 (Testes de software).
- 2.2.7. O plano de validação deve ser elaborado por quem desenvolveu o software. No caso de softwares adquiridos de terceiros, o plano deverá ser elaborado pelo usuário final e que tenha o conhecimento do objetivo deste software.

3. Controle de usuário e acesso

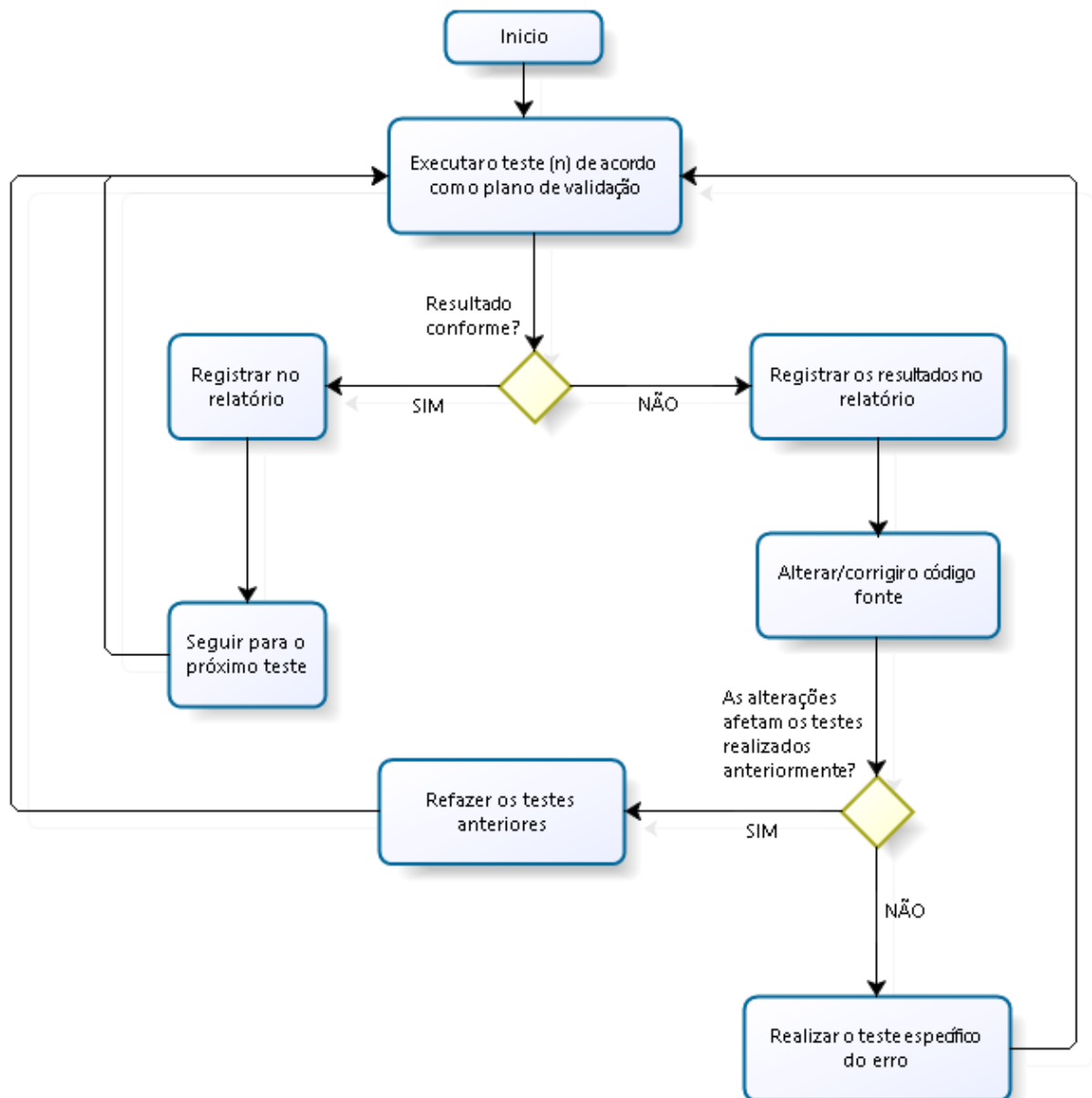
- 3.1. Deve existir no software uma forma de controle de acesso ao mesmo, garantindo restrição às informações e a confidencialidade das mesmas.
- 3.2. O controle de usuário deverá ser relacionado a uma identificação única e inequívoca do usuário, caracterizando assim sua assinatura eletrônica.
- 3.3. Apenas o pessoal autorizado pode ter acesso aos registros eletrônicos para a realização dos testes dos softwares.
- 3.4. Devem ser executados testes que provem a eficácia do sistema em impedir o acesso ao software de usuários não cadastrados e acessos de usuários às áreas do programa que não lhes dizem respeito.
- 3.5. Todas as ações (tais como modificações, inserções e deleções) devem ser identificáveis aos autores das mesmas e quando ocorrerem tais ações. Em alguns casos (em ensaios pertencentes à BPL, por exemplo) deve ser também incluída a justificativa para a mudança nos registros.
- 3.6. Devem ser realizados testes para verificar se a identidade do usuário que executou o acesso ao software é correspondente a quem fez as ações dentro do mesmo.

4. Testes de software

4.1. Generalidades

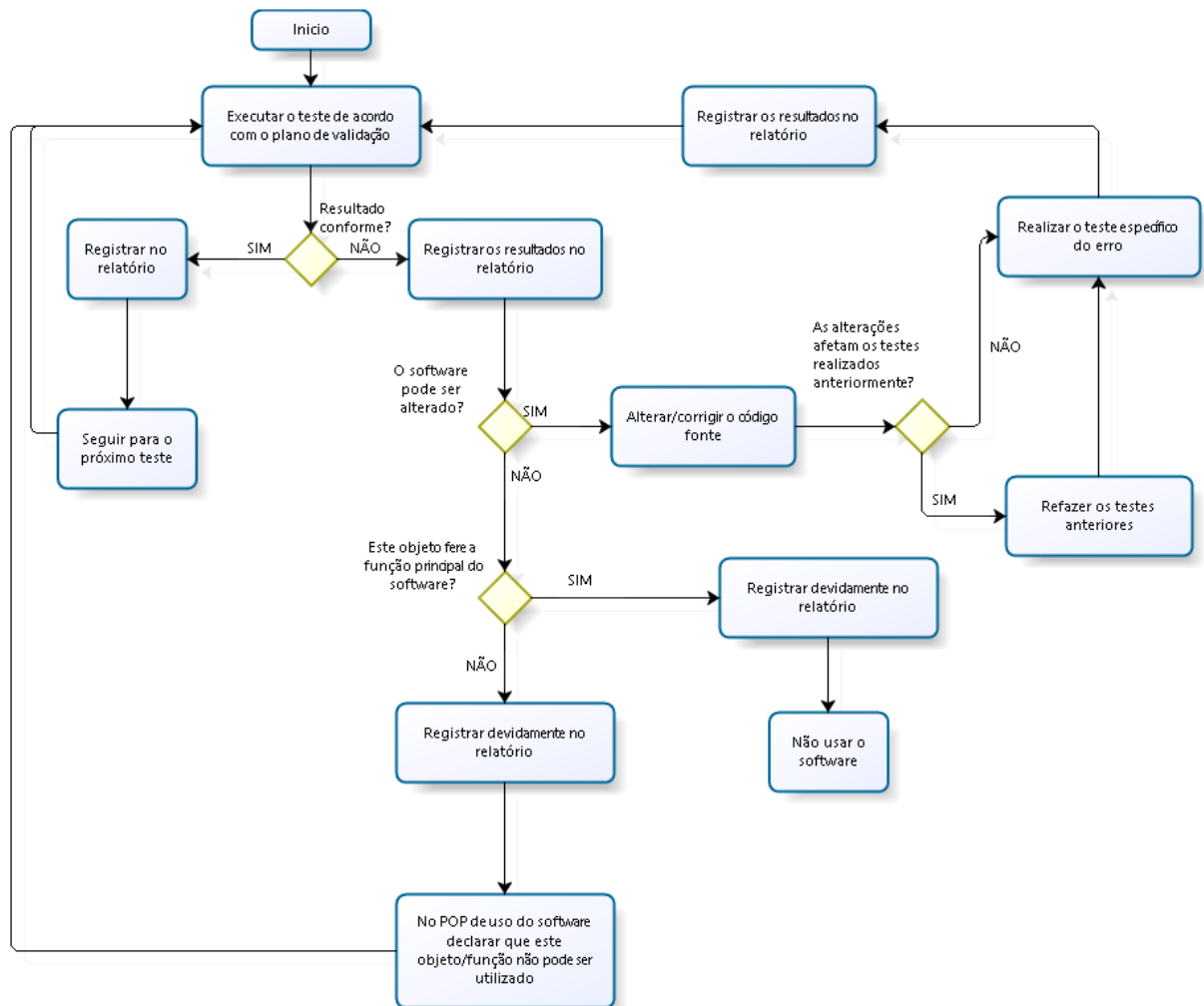
- 4.1.1. Os testes devem ser realizados, sempre que possível, por pessoas que não o desenvolveram, desde que possuam conhecimentos para a realização dos mesmos.
- 4.1.2. Os testes devem ser específicos para cada tipo de atividade executada pelo software (por exemplo: cálculo, inserção no banco de dados, o uso de condicionais).
- 4.1.3. Todos os testes realizados devem ser executados de acordo com o definido no plano de validação e registrados de forma clara e objetiva no relatório de validação (Figura 5 e Figura 6)
- 4.1.4. Após os testes, devem ser registradas as respectivas evidências, suas constatações e a conclusão, definindo se o presente software poderá ser total ou parcialmente aprovado, ou ainda, não poderá ser utilizado em hipótese alguma.

Figura 5 - Fluxograma de execução de testes para softwares desenvolvidos pelo laboratório



Fonte: Autoria própria

Figura 6 - Fluxograma de testes para softwares adquiridos de terceiros



Fonte: Autoria própria

- 4.1.5. Caso os resultados dos testes sejam não conformes, estes devem ser registrados no relatório de validação e deverão ser feitas as devidas correções no código fonte. Se estas alterações afetarem os testes realizados anteriormente, devem-se refazer os testes após a alteração. Caso contrário, deve-se fazer apenas o teste do respectivo erro.
- 4.1.6. Caso o software seja aprovado parcialmente, ou seja, apenas algumas funções poderão ser utilizadas, no POP de uso do software deverão ser definidas quais funções não poderão ser executadas.
- 4.1.7. Para softwares adquiridos de terceiros, os testes também devem ser executados segundo o plano de validação, sendo que todas as evidências devem ser igualmente registradas no relatório de validação. O fluxograma apresentado na Figura 6 mostra como devem ser efetuados os testes para softwares adquiridos de terceiros e as respectivas ações.
- 4.1.8. Se os resultados obtidos no teste de validação para softwares adquiridos de terceiros forem não conformes e o software puder ser alterado/corrigido de alguma forma, o responsável ou pessoa designada deve realizar as devidas modificações no software e refazer os testes (anteriores e/ou específico para o erro encontrado de acordo com a necessidade).
- 4.1.9. Caso o resultado obtido seja não conforme e não podendo ser alterado o software, deve-se analisar se este objeto fere ou não a atividade principal do programa. Caso positivo, registrar no relatório de validação e não utilizar o software.
- 4.1.10. Caso a função ou objeto que apresentou resultados não conformes com relação aos testes, porém esta não conformidade não fere a finalidade do software e não compromete nenhum outro objeto/função, deve-se realizar os devidos registros no relatório e declarar no POP de uso do software que este objeto/função não pode ser utilizado.

4.2. Objetos ou funções que realizam cálculos

- 4.2.1. São considerados para efeito destes testes as funções e objetos que realizam cálculos numéricos, com datas ou valores. Operadores Lógicos (e, ou, não...) são tratados no item 4.4 (Objetos que fazem uso de condicionais).
- 4.2.2. Deve-se realizar um teste funcional, fornecendo valores conhecidos ao software, objeto ou função, obtendo os resultados e comparando os dados de saída com o determinado manualmente. Ambos devem ser registrados no relatório de validação (valor recebido, através de captura de tela, e valor calculado experimentalmente).
- 4.2.3. Deverão ser utilizados valores limites nos testes considerando os tipos de variáveis (decimal, inteiro, booleano...) definidos no código-fonte para softwares desenvolvidos pelo laboratório. Para softwares adquiridos de terceiro, usar valores compreendidos dentro de uma escala do uso em rotina do software no laboratório.

Nota: Podem ser utilizadas planilhas de cálculos para a verificação dos resultados obtidos do software somente se estas já tiverem sido validadas anteriormente.

4.3. Objetos relacionados a banco de dados

- 4.3.1. São considerados para efeito deste teste, funções e objetos relacionados a atividades com banco de dados (inserção, deleção, alteração, pesquisa de informações) de qualquer natureza, seja trabalhando com números, caracteres, textos ou arquivos.
- 4.3.2. Quando é executada alguma forma de pesquisa do banco de dados, deve-se pesquisar informações existentes e inexistentes. Avaliar se a resposta recebida é correspondente ao que foi pesquisado. Considerar pesquisas de similares para que não ocorram falsos positivos. Considerar respostas únicas e múltiplas.

4.3.3. Devem-se selecionar informações a serem inseridas no banco de dados do software e avaliar o comportamento frente às atividades executadas. Verificar se as mesmas fazem parte da base de dados após a inserção, e se constam nos devidos registros.

Nota: podem ser incluídas novas informações (inexistentes) na base de dados ou, se aplicável, realizar testes de redundância (inserir informações que já existam: por exemplo, tentar inserir outro usuário com a mesma identificação para verificar o tratamento da situação)

4.3.4. Após a inserção deverá ser verificado se a informação contém a identificação dos responsáveis por esta atividade e a respectiva data. Realizar estes testes com diferentes usuários e, quando aplicável, de diferentes níveis de acesso.

4.3.5. Devem ser realizados testes de exclusão de informações existente na base de dados e avaliar o comportamento do software. As informações deletadas não devem ser perdidas definitivamente a fim de se garantir o histórico de alterações.

4.3.6. Deve-se alterar informações existentes no banco de dados do software e analisar como o programa se comporta frente a estas modificações. O software deve armazenar a nova informação sem perder a informação anterior (modificada).

Nota: Por exemplo, para informações excluídas e/ou alteradas podem ser utilizados banco de dados contendo as informações históricas dos registros ou mesmo outras tabelas dentro do próprio banco de dados.

4.3.7. Após qualquer deleção ou alteração de dados, o sistema deve apresentar a identificação do responsável pela deleção ou alteração, assim como a respectiva data da ação. Em alguns casos (em ensaios pertencentes à BPL, por exemplo) deve ser também incluída a justificativa para a mudança nos registros.

4.3.8. Deve-se registrar no relatório de validação, todos os testes realizados de inserção, deleção, alteração e pesquisa e respectivos resultados.

4.3.9. No caso de se utilizar programação diretamente no banco de dados, os mesmos testes devem ser aplicados, considerando que se trata também de um código-fonte criado.

4.4. Objetos que fazem uso de condicionais

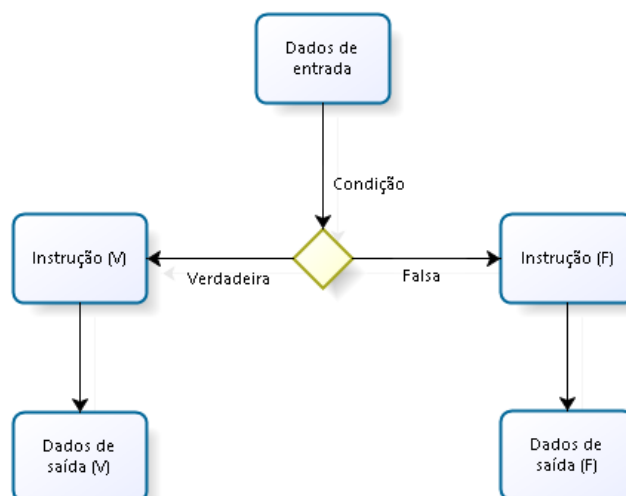
4.4.1. São considerados para efeito deste tipo de teste objetos e funções que fazem o uso de condicionais, levando em conta condições de “verdadeiro” ou “falso”, sendo que para cada caso, o software deve seguir instruções diferentes.

4.4.2. Testar as condições definidas pelo software, inserindo dados que resultem em estados verdadeiros e falsos, avaliando se as devidas instruções consequentes são realizadas com êxito, comparando os dados de saída com os esperados, como mostrado no fluxograma da Figura 7.

4.4.3. Usar valores limítrofes para as condições em análise.

4.4.4. Registrar no relatório final os testes realizados, bem como os valores utilizados para estes.

Figura 7 - Fluxograma de teste de condicionais



Fonte: Autoria própria

5. Testes de planilhas eletrônicas

5.1. Generalidades

- 5.1.1. São considerados para efeito destes testes, os arquivos (desenvolvidos pelo próprio laboratório ou adquiridos de terceiros) gerados ou editados, através de programas de computador que fazem o uso de tabelas (compostas por células) para realização de cálculos numéricos com datas, valores e apresentação de dados (podendo ser: numéricos, gráficos e literais),
- 5.1.2. A validação deve ser realizada por célula ou por blocos de células que contenham fórmulas a serem aplicadas a outras células ou outros blocos.
- 5.1.3. Deve-se proteger as células que contenham fórmulas. Células que contenham dados que não podem ser alterados (por exemplo, constantes utilizadas em funções) devem ser igualmente protegidas para impedir a entrada de valores numéricos e/ou textuais equivocados, entradas indesejadas no campo da planilha ou provocando alterações nas fórmulas.
- 5.1.4. Se a função a ser utilizada está salva em uma 'Macro' (no caso de planilhas eletrônicas que fazem o uso do VBA – Visual Basic for Applications) deve-se

validar a macro considerando a validação como descrita no item 4 (Testes de software), por se tratar de um código-fonte.

5.1.5. Para planilhas eletrônicas que fazem o uso da linguagem de programação 'Python', considerar a validação como descrita no item 4 (Testes de software) por se tratar de um código-fonte.

5.1.6. Todos os resultados devem ser devidamente registrados no relatório de validação e devem ser datados e assinados pelo responsável que realizou os respectivos testes.

5.2. Para funções que realizam cálculos

5.2.1. São consideradas para efeito deste teste as funções que realizam cálculos numéricos, com datas ou valores, sejam eles de qualquer tipo.

5.2.2. Deve-se verificar se a função utilizada está correta, inserindo dados numéricos na planilha, cujo resultado é conhecido através de cálculos executados manualmente. Executar a função, quando aplicável, e analisar se o resultado obtido pela planilha é o esperado.

5.2.3. A validação dos resultados dos cálculos numéricos realizados pelas planilhas eletrônicas deve ser devidamente registrada, apresentando os resultados e as fórmulas avaliadas.

5.3. Para funções condicionais

5.3.1. São consideradas para efeito deste tipo de teste as funções que fazem o uso de condicionais. Deve-se levar em conta quando da execução dos testes que, para cada condição, a célula (ou bloco de células) deve obedecer às diferentes instruções contemplando todos os possíveis resultados do acumulado das condicionais.

5.3.2. Inserir dados na célula ou no bloco de células (que são referenciados pela fórmula da condicional) com valores de saída (respostas) previamente conhecidos, executar a função e avaliar se a resposta obtida é a esperada (ver Figura 7).

Nota: Da mesma maneira, se uma função condicional estiver inserida em uma macro, deve-se inserir dados, executar a macro e analisar se as ações subsequentes são realizadas com êxito.

6. Manutenção/Backup

- 6.1. O laboratório deve possuir procedimentos estabelecidos de manutenção e atualização dos softwares por ele desenvolvido ou obtido de terceiros, as quais devem ser realizadas por pessoal qualificado e autorizado para esta atividade.
- 6.2. Nas atividades que causem alterações no software (correção ou atualização, por exemplo), o mesmo deve ser validado novamente. A extensão desta nova validação pode ser total ou parcial dependendo de quais funções/objetos foram alterados.
- 6.3. A cada nova validação são necessários novos planos de validação, registros e relatórios de validação. Estes devem ser arquivados junto dos documentos das validações anteriores os quais geram um histórico dos testes realizados em cada versão do software.
- 6.4. Devem existir procedimentos para a reinstalação do software que assegure que nenhum registro seja perdido.
- 6.5. O laboratório deve possuir cópias de segurança dos códigos-fonte de cada software por ele desenvolvido, garantindo também um histórico destes códigos, permitindo a qualquer momento retomar uma versão anterior.
- 6.6. No caso de softwares adquiridos de terceiros deve existir procedimentos que garantam o monitoramento de correções lançadas pelo o fabricante e que estas sejam aplicadas o mais rápido possível. Pode ser necessário que sejam realizados novos testes e, por consequência, nova validação.

7. Relatório de validação e demais registros

- 7.1. Deve constar nos registros a identificação do responsável pelo teste, assim como a data de realização do mesmo.
- 7.2. Deve-se manter todos os registros derivados dos testes dos softwares enquanto o software estiver em uso.
- 7.3. Todos os resultados obtidos de cada teste definido pelo plano de validação devem ser registrados de forma clara e objetiva em um relatório de validação, incluindo a evidência de cada teste realizado, os resultados alcançados e sua conformidade ou não.
- 7.4. Devem-se incluir no relatório de resultados dos testes, pelo menos, os seguintes dados:
 - a) Título do relatório e identificação unívoca do software e a respectiva versão avaliada;
 - b) Identificação do(s) responsável(is) pelo teste;
 - c) Data efetiva do início da utilização do software.

Nota: O software somente estará apto a ser utilizado após a conclusão da validação, emissão do relatório de validação e do respectivo procedimento de uso.

- d) Descrição do software, contendo seu objetivo principal, as funções essenciais do mesmo e os requisitos mínimos de hardware e software para sua utilização
- e) Relação dos testes realizados com os que foram descritos no plano de validação; se por algum motivo algum teste não foi realizado ou foi modificado descrever aqui a devida justificativa.
- f) Resultados obtidos e conclusão final da utilização do software (se este pode ser utilizado total ou parcialmente, ou ainda, não pode ser utilizado).

TUTORIAL TÉCNICO DE AVALIAÇÃO DA VALIDAÇÃO DE SOFTWARE PARA AVALIADORES/AUDITORES

1. Generalidades

Nota: 1. Para simplificar a denominação, os avaliadores (considerados externos ao laboratório) e auditores (normalmente responsáveis pelas auditorias internas) serão denominados como avaliadores apenas.

2. Este tutorial não é mandatório, nem os testes aqui mencionados são únicos e exclusivos. Este tutorial é apenas uma abordagem do roteiro apresentado, servindo de base para a auditoria. Assim, cabe ao avaliador definir quais os tipos de testes que irá realizar no software em estudo, porém é de suma importância que o mesmo aborde todos os itens do roteiro e seus requisitos.

- 1.1. O avaliador é responsável por verificar os softwares do laboratório, de maneira amostral, que afetem direta ou indiretamente a qualidade, garantindo que estes estejam executando sua tarefa com êxito e assegurando que os softwares sejam validados e mantidos em conformidade.
- 1.2. O avaliador deve garantir que o software esteja funcionando corretamente, executando alguns testes no mesmo, comparando os resultados com o que foi descrito no relatório realizado pelo desenvolvedor. Poderá executar qualquer teste que julgar necessário, mesmo que não conste no relatório de validação, mas que seja conveniente para a busca de evidências que comprovem a funcionalidade do software.
- 1.3. O avaliador deve analisar documentos com o intuito de determinar as conformidades do sistema. Estes documentos podem incluir, mas não se limitar, a registros descritos pelo desenvolvedor, procedimentos operacionais padrão, relatórios de validação (atuais e de versões anteriores do software) e relatórios de auditorias anteriores.

2. Requisitos gerais

2.1. Procedimentos Operacionais Padrão para validação de softwares

2.1.1. Deve-se avaliar se existem procedimentos operacionais padrão (POP) para validação tanto para os softwares desenvolvidos pelo laboratório quanto para os adquiridos de terceiros. Avaliar se também existem POPs de uso do software.

2.2. Plano de validação

2.2.1. Deve-se avaliar se o software em análise possui um plano de validação, assegurando que atende ao uso pretendido. Analisar se consta no plano de validação:

- a) Todos os testes realizados no software;
- b) As informações mínimas;
- c) Os procedimentos de controle de mudança de versões.

3. Controle de usuário e acesso

3.1. Deve-se analisar se existe um controle de acesso de usuário e se este é restrito e confidencial. Assegurar que todas as ações realizadas no software são identificáveis aos autores corretos, para isso deve-se realizar testes para a confirmação da assinatura eletrônica:

- a) Fazer login com um usuário existente;
- b) Executar algum tipo de ação no software (inserção, deleção, alteração de dados);
- c) Avaliar se o usuário que executou tal ação é o mesmo que consta nos registros do programa.

4. Testes de softwares

4.1. Generalidades

4.1.1. Deve-se testar os softwares, de maneira amostral, sejam os desenvolvidos pelo próprio laboratório quanto os adquiridos de terceiros. Cada teste deve ser específico para cada tipo de função executada pelo software (cálculo, relacionados ao banco de dados e que fazem o uso de condicionais).

4.2. Objetos ou funções que realizam cálculos

4.2.1. Deve-se avaliar os objetos e funções dos softwares que realizam cálculos. Para isso devem-se realizar alguns testes verificando se o software está funcionando de maneira adequada.

- a) Inserir um ou mais números (valores limites) no software;
- b) Executar a função matemática, cuja resposta seja conhecida;
- c) Avaliar se o resultado obtido é o esperado. Caso o resultado obtido não seja o esperado, tem-se uma não conformidade.

4.3. Objetos relacionados a banco de dados

4.3.1. Deve-se avaliar os objetos e funções dos softwares que estão relacionados com banco de dados. Realizar alguns testes (inserção, deleção, alteração, pesquisa de informações) com o objetivo de verificar se o programa funciona conforme o uso pretendido.

4.3.2. Para pesquisas de dados:

- a) Pesquisar uma informação existente no banco de dados;
- b) Verificar se a resposta obtida pela pesquisa corresponde a esperada;
- c) Pesquisar uma informação inexistente no banco de dados;
- d) Esperar alguma mensagem de erro, informando que a informação pesquisada não consta nos registros.

4.3.3. Para inserção de novos dados:

- a) Inserir uma nova informação no banco de dados do software e salvá-la;
- b) Pesquisar a informação previamente inserida;
- c) Verificar se esta foi devidamente salva no banco de dados. Caso isso não ocorra, considerar como um erro do software.

4.3.4. Para inserção de dados existentes:

- a) Inserir uma informação já existente no banco de dados do software (com o mesmo identificador, por exemplo, uma amostra com o mesmo código, um equipamento com o mesmo identificador, ou seja, verificar o uso correto de chaves primárias e estrangeiras);
- b) Esperar que apareça uma mensagem de erro, informando que este dado já existe. Caso isso não ocorra, considerar como uma não conformidade.

4.3.5. Para exclusão de dados:

- a) Apagar uma informação existente no banco de dados do software;
- b) Pesquisar a informação previamente deletada;
- c) Esperar uma mensagem informando que esta informação (recém-apagada) não consta mais no banco de dados do sistema. Caso isso não aconteça, definir essa função como uma não conformidade;
- d) Pesquisar a informação previamente apagada em um banco de dados ou tabela, onde ficam todos os registros apagados e alterados do software;
- e) Esperar que a esta informação esteja nesses registros. Se não estiver, caracterizar essa função como não conforme.

4.3.6. Para alteração de dados:

- a) Alterar uma informação existente no banco de dados e salvá-la;
- b) Pesquisar a nova informação, previamente alterada;
- c) Verificar se esta foi devidamente alterada e salva no sistema.

4.3.7. Todas as alterações realizadas pelo avaliador devem ser revertidas ao final da auditoria.

4.4. Objetos que fazem o uso de condicionais

4.4.1. Devem-se avaliar os objetos e funções dos softwares que fazem o uso de condicionais e operadores lógicos. Realizar alguns testes descritos no roteiro, verificando a veracidade dos dados obtidos.

- a) Inserir dados que resultem em uma condição conhecida (por exemplo, o uso de valores que façam com que uma amostra seja aprovada e outro valor com que a amostra seja tida como reprovada);
- b) Executar a condição definida pelo software;
- c) Avaliar se o resultado obtido é o resultado esperado. Caso o software realize uma condição diferente da esperada, assumir um erro;
- d) Repetir o processo com outros dados, testando condições e casos diferentes.

5. Testes de planilhas eletrônicas

5.1. Generalidades

5.1.1. Deve-se avaliar as planilhas eletrônicas, de maneira amostral, usadas pelo laboratório (desenvolvidas pelo laboratório ou adquiridas de terceiros), realizando os devidos testes de cálculos e de condicionais.

5.2. Para funções que realizam cálculos:

- a) Inserir um valor numérico na célula da planilha que contenha a função que se deseja validar;
- b) Executar a função;

- c) Avaliar o resultado e comparar se o resultado obtido pela planilha é o resultado esperado. Caso for um resultado diferente, assumir como sendo um erro na planilha.

5.3. Para funções condicionais:

- a) Inserir um dado numérico na célula, que contenha a fórmula da condicional que se deseja validar;
- b) Executar a função na célula;
- c) Avaliar se a resposta obtida pela planilha é a resposta esperada. Alterar o valor para que a condicional resulte em uma condição diferente (por exemplo, o uso de valores que façam com que uma amostra seja aprovada e outro valor com que a amostra seja tida como reprovada). Caso não for a resposta esperada, assumir como sendo um erro na planilha.

6. Manutenção/Backup

- 6.1. Deve-se analisar se o laboratório possui procedimentos de manutenção e atualização dos softwares por ele desenvolvido ou adquirido de terceiros. Avaliar também a cópia de segurança dos registros eletrônicos por ele armazenado, de acordo com o item 6 do roteiro.
- 6.2. Deve-se analisar se existe no laboratório cópias de segurança e um histórico dos códigos-fonte de todos os softwares desenvolvidos por ele, bem como a cópia de segurança dos códigos-fontes dos mesmos. E no caso dos softwares adquiridos de terceiros, deve-se analisar se existem procedimentos de monitoramento de correções lançadas pelo fabricante.

7. Relatórios e demais registros

- 7.1. Devem-se analisar os relatórios de validação em busca de evidências e possíveis não conformidades nos mesmos. Deve-se verificar, no mínimo, se:
- a) Os registros dos testes de software estão documentados, para prover evidência da conformidade do sistema de gestão de qualidade;
 - b) As versões pertinentes (com suas devidas alterações e atualizações) dos documentos estão disponíveis no local de uso, junto com a versão anterior;
 - c) Os documentos são legíveis e identificáveis;
 - d) Existe procedimento para definir os controles necessários para identificação, armazenamento e recuperação dos registros.

8. Constatações

- 8.1. Registrar todos os resultados e constatações da análise dos softwares auditados (tanto os desenvolvidos pelo laboratório, quanto os adquiridos de terceiros) concluindo, de acordo com as evidências, se os requisitos das normas foram atendidos, ou seja, gerando uma conclusão no relatório de auditoria nos respectivos itens de acordo com o acúmulo das constatações.

Conclusão

O presente trabalho fundamenta que o conceito de “qualidade de software” é algo subjetivo e relativo, uma vez que um software que atenda a norma de qualidade de produto de software, isto é, possui as características propostas pela norma ISO/IEC 25010:2011 é tido como um software de qualidade, entretanto não necessariamente este produto vai atender as particularidades do sistema de gestão de qualidade para laboratórios

Tendo isso em mente, a criação de um roteiro de apoio à validação de softwares para laboratórios irá complementar o que a norma de qualidade de software propõe, isto é, além do software atender as características impostas por esta norma, os softwares presentes no laboratório, tanto os desenvolvidos pelo laboratório, quanto os adquiridos de terceiros, devem ser testados e validados de maneira a atender as especificidades da ISO/IEC 17025.

É importante ressaltar que mesmo o que é solicitado pelas normas de gestão da qualidade para laboratórios não é feito de maneira explícita. Assim o presente roteiro complementa a forma como devem ser atendidos os requisitos, assim como é feito com os guias de validação de métodos que é específico para cada área, podendo ser mais ou menos exigente.

O roteiro descreve que os testes desenvolvidos devem ser específicos para cada tipo de atividade realizada pelos softwares (cálculos, relacionadas a banco de dados e uso de condicionais), incluindo também os testes para planilhas eletrônicas.

Além disso, o presente trabalho ainda contempla um tutorial técnico de apoio para auxiliar o avaliador na tarefa de verificar o software e sua respectiva validação, considerando que este pode não possuir conhecimentos prévios de programação. Este tutorial demonstra minimamente como o auditor pode seguir na busca de evidências e não conformidades nos softwares presentes no laboratório.

Conclui-se, portanto, que o roteiro de apoio à validação de software, o qual auxilia o desenvolvedor do software, acoplado ao tutorial técnico de apoio ao avaliador são ferramentas paralelas aqui desenvolvidas às normas de qualidade de software, podendo ser empregadas nas mais diferentes áreas e propósitos relacionados aos sistemas de gestão da qualidade para laboratórios.

Referências bibliográficas

- [1] OLIVARES, I.R.B. **Gestão de Qualidade em Laboratórios**, 2. Ed., revisada e ampliada. Campinas: Átomo, 2009. 146p.
- [2] ISO. About ISO. Disponível em: <<http://www.iso.org/iso/home/about.htm>> Acesso em: 19 fev. 2016.
- [3] CAMACHO, M. S. **A importância de um certificado ISO para uma organização**. 2013. 43f. Tese (Pós graduação "Lato-Sensu") – AVM Faculdade integrada, Rio de Janeiro. 2013.
- [4] GROCHAU, I. H., TEN CATEN, C. S. **A process approach to ISO/IEC 17025 in the implementation of a quality management system in testing laboratories**. Accreditation Quality Assurance. v. 17, P. 519-527. 2012.
- [5] ÖRNEMARK, U. et al. **Policies, requirements and surveys concerning frequency for participation in proficiency testing schemes**. Accreditation and Quality Assurance, v. 9, n. 10-11, p. 729-732. 2004.
- [6] CORTEZ, L. et al. **How to interpret information from proficiency test exercises concerning the relative performance of accredited laboratories**. Accreditation and Quality Assurance, v. 8, n. 11, p. 511-513. 2003.
- [7] MARTINS, R. A.; NETO, P. L. O. C. **Indicadores de desempenho para a gestão pela qualidade total: uma proposta de sistematização**. Gestão e produção. V. 5, n. 3, p. 298-311, 1998.
- [8] ORLANDI, Leandro. **Sistemas de Informação ou Sistemas Informatizados?**. São Paulo. 2007,1p. Disponível em: <http://www.bonde.com.br/?id_bonde=1-14--2789-20070605> Acesso em: 25 fev. 2016
- [9] COSTAL, G. C. S. Z.; MARTINS, R. A. **Dificuldades na informatização de um sistema de gestão da qualidade baseado na norma ABNT ISO 9001**. XXXI Encontro Nacional de Engenharia de Produção (ENEGEP). Belo Horizonte, 2011.
- [10] NAZÁRIO, P. Aulas Faccamp (faculdade Campo Limpo paulista) – **A importância de sistemas de informação para a competitividade logística**. Disponível em: <<http://www.tecspace.com.br/paginas/aula/faccamp/TI/Texto04.pdf>>. Acesso em: 06 dez. 2015.
- [11] PEREIRA, M. F. F. **Gerenciamento da informação: um diagnóstico da micro e pequena empresa industrial de Londrina**. 2003. 154f. Tese (Mestrado Engenharia de produção) – Universidade Federal de Santa Catarina, Florianópolis, 2003.
- [12] BERALDI, I. C.; FILHO, E. E. **Impacto da tecnologia de informação na gestão de pequenas empresas**. Ci. Inf., Brasília, v. 29, n. 1, p 46-50, 2000.
- [13] QUINET, M. **Fundamentos de arquitetura de computadores**. Universidade Federal Fluminense. 58p.
- [14] KALINOWSKI, M. Introdução à Inspeção de Software. **Revista Engenharia de Software: Qualidade de software**, 1. Ed. Artigo de Capa. 2008. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-inspecao-de-software/8037>>. Acesso em: 12 dez. de 2015

- [15] KOSCIANSKI, A.; SOARES, M. S., **Qualidade de software – Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. 2. Ed. Novatec, 2007. 395p.
- [16] RAMOS, R. A. **Introdução a verificação, validação e teste de software**. Engenharia de software. 2012 (Baseado na apresentação do LABS – ICMC-USP). Disponível em: <<http://www.labes.icmc.usp.br>>. Acesso em: 08 jan. 2016.
- [17] PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**. McGraw-Hill, 6. Ed, Nova York, NY, 2005. Disponível na Revista Engenharia de Software – Qualidade de Software <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>> Acesso em: 25 fev. 2016.
- [18] CLAUDIO, A. Engenharia de Software - Introdução a Teste de Software. **Engenharia De software: Qualidade de software**. 1. Ed. 2008. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035#ixzz44xcT7NOF>> Acesso em 05 abr. 2016
- [19] MYERS, G. **The art of software testing**. New Jersey. Wiley. 3. Ed., 2012. 254p
- [20] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 17025**. Requisitos gerais para a competência de laboratórios de ensaio e calibração. Rio de Janeiro, 2005. 31p.
- [21] INMETRO – INSTITUTO NACIONAL DE METROLOGIA, QUALIDADE E TECNOLOGIA. **NIT-DICLA-035**. Princípios das boas práticas de laboratório - BPL. 2011. 19p.
- [22] INMETRO – INSTITUTO NACIONAL DE METROLOGIA, QUALIDADE E TECNOLOGIA. **NIT-DICLA-038**. A aplicação dos princípios da BPL aos sistemas informatizados. 2011. 13p.
- [23] ISO/IEC 25010:2011. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. EUA. 2011. 34p.
- [24] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9000**. Sistemas de gestão de qualidade – Fundamentos e vocabulário. Rio de Janeiro, 2005. 35p.