



VII SINGEP

Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade
International Symposium on Project Management, Innovation and Sustainability

ISSN: 2317-8302

GERENCIAMENTO DE LIBERAÇÃO DE VERSÃO COM SISTEMA ESPECIALISTA NA CLASSIFICAÇÃO DE CRITICIDADE DE VERSÃO DE SOFTWARE

DACYR DANTE DE OLIVEIRA GATTO

Universidade Nove de Julho

ROSANA CORDOVIL DA SILVA

UNINOVE

RENATO JOSÉ SASSI

UNINOVE – Universidade Nove de Julho



GERENCIAMENTO DE LIBERAÇÃO DE VERSÃO COM SISTEMA ESPECIALISTA NA CLASSIFICAÇÃO DE CRITICIDADE DE VERSÃO DE SOFTWARE

Resumo

O gerenciamento de serviços de Tecnologia da Informação (TI) visa garantir a entrega de serviços de TI. Metodologias e boas práticas apoiam essa demanda. Dentre as metodologias destacam-se o *Business Process Management* e o *Six Sigma* e entre as boas práticas destacam-se a *Information Technology Infrastructure Library* e o *Project Management Body of Knowledge*. A aplicação destas metodologias e boas práticas permite a definição e execução do processo de gerenciamento de liberação de versão de *software*. O aumento da complexidade das versões e a subjetividade na sua classificação gera a necessidade de apoiar a classificação das versões de modo objetivo. Buscando reduzir a subjetividade no processo aplica-se uma técnica Sistema Especialista, no processo de gerenciamento de liberação de versão. O objetivo deste artigo foi classificar a criticidade de versão de *software* com apoio de Sistema Especialista para reduzir a subjetividade na execução do processo. A metodologia de pesquisa adotada neste trabalho foi definida como documental e experimental. A condução da pesquisa foi dividida em cinco Fases, sendo as Fases um a quatro, pesquisa documental e a Fase cinco, pesquisa experimental. Os resultados obtidos atendem o critério de criticidade antes conhecidos apenas pelos especialistas, diminuindo a subjetividade na classificação.

Palavras-chave: Processo de gerenciamento de liberação de versão de *software*, BPM, Six Sigma, ITIL, PMBOK, Sistema Especialista

Abstract

The management of Information Technology (IT) services aims to guarantee the delivery of IT services. Methodologies and good practices support this demand. Among the methodologies are Business Process Management and Six Sigma, and the best practices include Information Technology Infrastructure Library and Project Management Body of Knowledge. The application of these methodologies and good practices allows the definition and execution of the software release management process. Increasing the complexity of the versions and the subjectivity in their classification generates the need to support the classification of the versions in an objective way. Seeking to reduce subjectivity in the process applies an Expert System technique in the software release management process. The purpose of this article was to classify software version criticality with Expert System support to reduce subjectivity in the execution of the process. The research methodology adopted in this work was defined as documentary and experimental. The conduction of the research was divided into five Phases, being Phases one to four, documentary research and Phase five, experimental research. The results obtained meet the criterion of criticality previously known only by the specialists, reducing the subjectivity in the classification.

Keywords: *Software Release Management Process, BPM, Six Sigma, ITIL, PMBOK, Expert System*



1 Introdução

Organizações tem usado o gerenciamento de serviços de Tecnologia da Informação (TI) como instrumento de gestão e controle do ambiente computacional, visando garantir a entrega de serviços de TI para atender as necessidades de negócio, gerenciando assim as tecnologias, pessoas e processos envolvidos, buscando vantagens competitivas em um mercado globalizado e em constante mudança. Para apoiar esta demanda foram desenvolvidos por empresas de pesquisa boas práticas e metodologias buscando melhorar as capacidades em serviços entregues pelas organizações (Axelos, 2013).

Metodologias podem ser definidas como sistema de métodos, princípios e regras que regulam uma determinada disciplina em ciências, já boas práticas definem-se como um conjunto das técnicas, processos, procedimentos e atividades identificados, utilizados, comprovados e reconhecidos por diversas organizações, em determinada área, como sendo os melhores quanto ao mérito, eficácia e sucesso alcançados pela sua aplicação na realização de uma tarefa. (Vitoriano & Neto, 2015).

Organizações de desenvolvimento de *software* têm, dentre os processos gerenciáveis de TI, o processo de gerenciamento de liberação de versão de *software*, que visa construir, testar e entregar *software* capaz de suportar as especificações solicitadas pelo cliente, e assim entregar os objetivos pretendidos (ITIL *Service Lifecycle Publication Suite*, 2013).

Dentre as metodologias mencionadas destacam-se o *Business Process Management* (BPM) (*Association of Business Process Management Professionals* [ABPMP], 2018) e *Six Sigma* (*Six Sigma Institute*, 2018), e entre as boas práticas destacam-se a *Information Technology Infrastructure Library* (ITIL) (Axelos, 2013) e o *Project Management Body of Knowledge* (PMBOK), (*Project Management Institute* [PMI], 2018).

Com a aplicação destas metodologias e boas práticas tem-se o desenho e a definição do processo de gerenciamento de liberação de versão de *software*, assim como a definição de indicadores de execução e de qualidade. Estes indicadores permitem o desenvolvimento de uma base de dados com todos os indicadores da execução do processo, sendo alimentada sempre que o processo de gerenciamento de liberação de versão de *software* é executado.

Durante a execução do processo classifica-se a criticidade de uma versão em Alta, Média e Baixa, utilizando-se os indicadores citados como referência. Com o aumento da complexidade das versões de *software* em relação ao impacto que estas podem causar ao negócio, surgiu a necessidade de apoiar a classificação das versões de *software* de modo formal, dentro de critérios padronizados e assim emitir um parecer de criticidade de cada versão antes que esta fosse liberada em produção, eliminando a falsa impressão que todas as versões de *software* liberadas são similares.

Os especialistas responsáveis pela execução do processo emitem um parecer ao durante a execução do processo, classificando a versão de *software*, tornando a análise de criticidade de versão uma atividade subjetiva ao entendimento de cada especialista, dificultado o processo de tomada de decisão da criticidade da versão de *software*.

Com o objetivo de reduzir a subjetividade na execução do processo de gerenciamento de liberação de versão de *software* e assim promover tomadas de decisão mais precisas pode-se aplicar uma técnica de Inteligência Artificial (IA) como, por exemplo, a técnica de Sistema Especialista (SE). (Liao, 2018).

As metodologias e boas práticas permitem o desenho e a definição do processo de gerenciamento de liberação de versão de *software*, assim como a definição dos seus indicadores de execução e de qualidade. Com o processo de gerenciamento de liberação de versão de *software* sendo executado com sucesso, e com a aplicação da técnica de IA, Sistemas Especialistas, pode-se apoiar à classificação da criticidade da versão de *software*.



2 Problema de Pesquisa e Objetivo

A aplicação das metodologias e boas práticas busca que o processo de gerenciamento de liberação de versão de *software* seja executado sem erros e falhas. Durante a execução do processo classifica-se a criticidade da versão em Alta; Média e Baixa.

Com o aumento da complexidade das versões em relação ao impacto que estas podem causar ao negócio, surgiu a necessidade de apoiar a classificação das versões de *software* de modo formal. O problema é que o especialista no domínio é quem atribui a classificação, porém de forma subjetiva e informal. Isto cria uma falsa impressão que todas as versões liberadas são similares para os demais envolvidos no processo.

Um SE pode apoiar na classificação de criticidade de versão de *software*, auxiliando na tomada de decisão, uma vez que ele lida com problemas complexos do mundo real que necessita da análise e interpretação de um especialista humano e soluciona estes problemas através do uso de um modelo computacional do raciocínio de um especialista humano, de forma a chegar às mesmas conclusões que este especialista chegaria, porém agora de modo formal e objetivo.

O objetivo deste artigo é a classificação da criticidade de versão de *software* com apoio de Sistema Especialista.

3 Fundamentação Teórica

3.1 Metodologias e Boas Práticas

Dentre as metodologias que auxiliam no mapeamento de processos de negócio, destaca-se a *Business Process Management* (BPM). A BPM é uma metodologia que auxilia a organização a compreender o seu negócio de forma clara, com uma abordagem disciplinada para identificar, desenhar, executar, documentar, medir, monitorar, controlar e melhorar processos automatizados ou não, a fim de alcançar os resultados pretendidos, consistentes e alinhados com as metas estratégicas de uma organização (Calazans, Kosloski & Guimarães, 2016).

Em relação as boas práticas, a ITIL destaca-se como um conjunto de boas práticas, que tem como objetivo ajustar as pessoas, os processos e a tecnologia visando o aumento da eficiência do gerenciamento de serviços de TI (ITIL *Service Lifecycle Publication Suite*, 2013).

O uso da ITIL torna os serviços de TI cada vez mais alinhados a necessidade do negócio, pois os prazos estabelecidos e a melhoria da qualidade estão diretamente relacionados com a melhoria do relacionamento com o negócio. Além disso, os serviços prestados são descritos de forma mais clara tornando maior o entendimento entre a TI e o negócio.

O alinhamento entre o negócio e a área de TI possibilita, a definição de critérios para a entrega do resultado gerado pelo processo, como por exemplo, a elaboração de indicadores, ou como são denominados na ITIL, *Key-Performance Indicators* (KPIs), proporcionando melhoria na medição dos processos, como também melhorias na comunicação, no gerenciamento de ativos e atividades profissionais desempenhadas. (Gatto, Sassi & Costa, 2017).

Com o foco na melhoria da qualidade de processos (sejam de negócios ou TI) e de seus indicadores, a metodologia *Six Sigma* vem ganhando a apreciação das organizações como uma abordagem para melhoria da qualidade com impacto positivo no desempenho do negócio. Essa consideração deu notoriedade não apenas na indústria de manufatura, mas



também em operações de serviços, mostrando que o *Six Sigma* está inserido no âmbito das discussões estratégicas das organizações. (*Six Sigma Institute*, 2018)

Quando foi difundido pela Motorola no final da década de 80, o *Six Sigma* teve como propósito ser uma iniciativa voltada para o controle da qualidade total, com ênfase na satisfação dos clientes e na eliminação de erros e falhas nos processos produtivos (Erdogan e Canatan, 2015).

A metodologia *Six Sigma* é um programa que trouxe contribuições em vários aspectos para incrementar as discussões sobre gestão da qualidade e gestão estratégica, a partir da percepção de que sua estrutura incluía requisitos essenciais como: conceito de qualidade; combinação das abordagens estatística e estratégica; sistematização metodológica para implementar projetos, formação de especialistas e gestão de projetos em empresas de todos os tamanhos. (Estorilio & Amitrano, 2013).

Segundo o PMI (2018), o PMBOK procura estruturar a estratégia para o gerenciamento de projetos, como por exemplo a implantação de metodologias ou boas práticas, disponibilizando processos e ferramentas para sua correta execução. (PMI, 2018).

O trinômio do gerenciamento de projetos é formado por Cronograma, Custos e Qualidade, a esse trinômio são acrescidas outras sete áreas que são escopo, recursos humanos, comunicação, riscos, aquisições, integração e partes interessadas, sendo esse conjunto chamado de as Dez Áreas do Conhecimento que são necessárias para se gerenciar um projeto.

Os benefícios do *Six Sigma* estão em garantir uma meta de desempenho e a redução da variabilidade dos processos de negócio, melhorando assim a sua qualidade, enquanto o PMBOK disponibiliza monitorações constantes da implantação e sua velocidade (Gatto, Farias e Sassi, 2017)

O PMBOK, portanto, oferece conhecimentos, habilidades e técnicas para a execução de projetos de forma efetiva e eficaz. Trata-se de uma competência estratégica para organizações, permitindo com que elas unam os resultados dos projetos com os objetivos do negócio (PMI, 2018).

3.2 Processo de Gerenciamento de Liberação de Versão

O processo de gerenciamento de liberação de versão é responsável pela implantação das mudanças aprovadas pelo gerenciamento de mudança, sejam estas de TI ou não-TI. Uma visão sistêmica sobre os serviços de TI favorece a proteção do ambiente e de seus serviços através de checagens e procedimentos formais, estruturados e definidos, e que todos os aspectos de uma liberação sejam considerados como um todo. Através deste gerenciamento, o ambiente fica protegido contra liberações que não obedecem aos procedimentos formais, pois estabelece mecanismos para isto (*ITIL Service Lifecycle Publication Suite*, 2013).

Dentre os objetivos pretendidos deste processo pode-se descrever:

- a) Definir e acordar com o cliente as diretrizes para o desenvolvimento de planos de liberação garantindo a compatibilidade de cada pacote a ser liberado;
- b) Garantir que a integridade de cada pacote de liberação seja mantida durante todo o processo;
- c) Garantir que todos os pacotes possam ser rastreados, instalados, testados, verificados e desinstalados se for o caso; registrar e gerenciar desvios, riscos e questões relacionados com os serviços novos ou alterados;
- d) Certificar que há transferência de conhecimento para permitir que o cliente e usuários possam aperfeiçoar os usos dos serviços para apoiar as atividades de seus negócios;
- e) Assegurar que as habilidades e o conhecimento necessários sejam transferidos para as operações e pessoal de apoio, permitindo manterem de forma eficaz e eficiente a entrega, a manutenção, e serviços de acordo com as garantias exigidas.



O processo de liberação de versão de *software* é composto pelos termos “Unidade de liberação” e “Pacote de liberação”, que são utilizados e definidos como demonstrado a seguir. (Fernandes & Abreu, 2014)

a) Unidade de Liberação: São partes de um *software* que são liberadas de acordo com a política de liberação adotada. Essas partes podem variar de acordo com o *software*, ou seja, pode ser liberado o sistema todo ou somente um módulo conforme demonstrado em destaque na Figura 1.

b) Pacote de Liberação: Conjunto de unidades de liberação ou até mesmo uma única unidade. Pode ser liberada uma nova versão de um *software* contendo o novo aplicativo, *script* de banco de dados e manuais de usuário, por exemplo. Esse conjunto é um pacote de liberação. A Figura 1 representa um pacote de liberação.

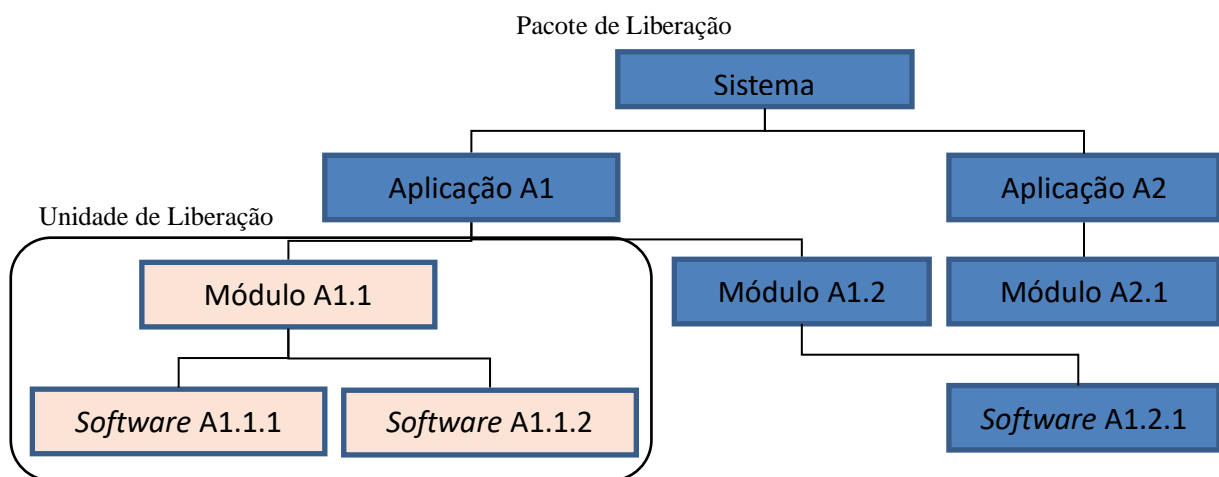


Figura 1: Unidade e Pacote de Liberação
Fonte: Autores (2018)

O processo de gerenciamento de liberação de versão tem também o objetivo de classificar a criticidade de uma unidade ou pacote de liberação antes que estes sejam colocados em produção. Cada unidade ou pacote de liberação podem conter componentes variados que, como citado anteriormente, podem causar impactos diversos quando implementados. (Barros, Salles, Gomes, Silva & Costa, 2015)

Torna-se necessário então um método para classificar esta criticidade e fornecer um parecer sobre a mesma, removendo a subjetividade daqueles que classifica. Uma ferramenta de apoio a decisão se torna importante para remover a subjetividade no processo de classificação, e assim torna-lo objetivo e preciso, como por exemplo um Sistema Especialista.

3.3 Sistema Especialista

Um Sistema Especialista (SE) é uma aplicação interativa da área da IA de apoio à tomada de decisão baseada em computador. Um SE pode ser entendido ainda como uma simulação computacional, que contém conhecimento, análise e habilidades extraídos de um ou mais especialistas humanos, em um domínio de problema específico. (Gupta & Singhal, 2013).

Para representar o desempenho de especialistas humanos, o SE deve possuir não só um conjunto de informações, mas também, a habilidade de utilizá-las na resolução de problemas.



Esta habilidade representa uma série de regras intuitivas que o especialista utiliza para resolver os problemas, e sua aplicação possibilita, de uma maneira mais econômica, a obtenção de soluções aceitáveis, embora nem sempre ótimas. (Adeli, 2014).

O principal objetivo de um SE é capturar o conhecimento de um especialista humano relativo a algum domínio específico e codificá-lo em um sistema computacional, de tal forma que o conhecimento capturado fique disponível para que usuários menos experientes possam resolver problemas do mesmo domínio (Wagner, 2017).

A estrutura básica para um SE é constituída por três elementos fundamentais: base de conhecimento, motor de inferência e interface com o usuário, como representado na Figura 2. (Lia, 2017).

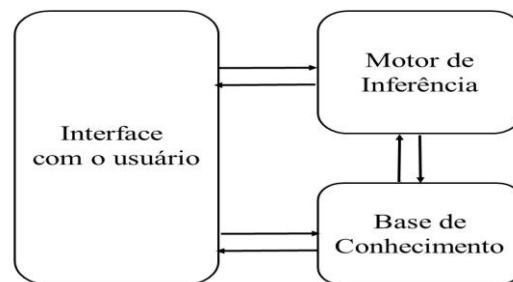


Figura 2: Componentes de um Sistema Especialista

Fonte: Autores (2018)

A tradicional base de dados com dados, arquivos, registros e seus relacionamentos estáticos são aqui substituídos por uma base de regras, fatos e heurísticas que correspondem ao conhecimento do especialista, ou dos especialistas do domínio sobre o qual foi construído o sistema. (Wagner, 2017).

O motor de inferência é o núcleo do sistema. É por intermédio dele que os fatos e regras e heurísticas que compõem a base de conhecimento são aplicadas no processo de resolução do problema. A capacidade do motor de inferência é baseada em uma combinação de procedimentos de raciocínios que se processam de forma regressiva e progressiva. (Lia, 2017).

Já a interface com o usuário, segundo Frizon, Silveira e Cunha (2015), é o elemento em que os desenvolvedores de SEs dedicam mais tempo projetando e implementando. É necessário que a interface com o usuário seja bastante flexível. Assim, a interação entre SE e usuário conduz um processo de navegação, eficiente, na base de conhecimento, durante o processamento das heurísticas.

Os SEs podem ser classificados de acordo com a classe de tarefas e/ou problemas para os quais são desenvolvidos (Frizon, Silveira & Cunha, 2015):

a) Interpretação: inferem descrições de situações a partir da observação de fatos, fazendo uma análise de dados e procurando determinar as relações entre eles e seus significados;

b) Diagnóstico: detectam falhas oriundas da interpretação de dados. Detectam os problemas mascarados por falhas dos equipamentos e falhas do próprio diagnóstico (que este não detectou por ter falhado);

c) Monitoramento: verificam, continuamente, um determinado comportamento em limites preestabelecidos;

d) Predição: a partir de uma modelagem de dados do passado e do presente, permite uma determinada previsão do futuro, baseando-se na análise de comportamento, fazendo uso



de raciocínios hipotéticos e verificando a tendência de acordo com a variação dos dados de entrada;

e) Planejamento: o sistema define um conjunto de ações a serem tomadas para se atingir um determinado objetivo. São estabelecidas etapas, subetapas e determinadas prioridades para as mesmas;

f) Projeto: é semelhante ao SE de planejamento; devem-se confeccionar especificações tais que sejam atendidos os objetivos dos requisitos particulares; esse tipo de SE é capaz de justificar a alternativa tomada para o projeto final e de fazer uso dessa justificativa para alternativas futuras;

g) Depuração: possuem mecanismos para fornecer soluções para o mau funcionamento provocado por distorções de dados;

h) Reparo: desenvolve e executa planos para administrar os reparos verificados na etapa de diagnóstico, seguindo um plano para administrar alguma solução encontrada em uma etapa de diagnóstico;

i) Instrução: tem mecanismos para verificar e corrigir o comportamento do aprendizado dos estudantes; SE empregado em ambientes educacionais;

j) Controle: governa o comportamento geral dos outros sistemas; é o mais complexo, pois deve interpretar os fatos de uma situação atual, verificando os dados passados e fazendo uma predição do futuro.

Wagner (2017) apresenta uma pesquisa realizada sobre 311 estudos de caso de uso de SEs entre 1984 e 2016 em diversas áreas de aplicação, dentre elas 14 cenários de aplicação de SEs em área de TI. Também é citado dentre os cenários estudados que 26 deles trabalhavam com aplicação de SEs em problemas de classificação. Farias e Sassi (2018) demonstram o uso de SE em um *Service Desk* baseado na ITIL, buscando melhorar a qualidade no atendimento ao cliente.

4 Metodologia

4.1 Caracterização Metodológica

Como referencial teórico para a elaboração deste artigo foram utilizadas literaturas (artigos de periódicos e congressos) referente as metodologias e boas práticas assim como referências de SEs para efeito de contextualização do conteúdo apresentado.

Os artigos de periódicos pesquisados foram obtidos do Portal de Periódicos CAPES/MEC (<http://www.periodicos.capes.gov.br/>) assim como dos Anais de Congressos relacionados aos temas, e as obras utilizadas são de autores relacionados ao referencial teórico da pesquisa.

A metodologia de pesquisa adotada neste trabalho foi definida como documental e experimental. A pesquisa bibliográfica foi elaborada a partir de documentos e bases de dados da empresa desenvolvedora de *software*, com a finalidade de descrever sistematicamente a situação e o problema encontrados, e investigar as possibilidades encontradas, buscando aplicar os conceitos teóricos apresentados no referencial elaborado.

A pesquisa documental refere-se ao procedimento da análise de documentos e dados que ainda não foram sistematizados e publicados, buscado encontrar informações e padrões em documentos ainda não tratados, pesquisar por relatórios, memorandos e bancos de dados relacionados à execução das atividades em uma empresa desenvolvedora de *software*.

A pesquisa experimental explica através da observação de experimentos, chegando a um consenso em que o investigador analisa e constrói hipóteses a respeito do problema, manipulando os fatores (Koche, 2003).



Percebeu-se com a realização da análise documental que a equipe de infraestrutura de aplicação alocada no cliente, encontrou na execução do processo de gerenciamento de liberação de versão de *software*, e na busca de orientações focadas em metodologias e boas práticas. A análise experimental focou em uma técnica de IA, no caso Sistemas Especialistas para apoiar à classificação da criticidade da versão de *software*.

A pesquisa documental foi realizada no ambiente da empresa Softplan (Softplan, 2018) entre 2013 e 2016 dentro do ambiente do seu principal cliente em São Paulo, que forneceu autorização formal para a pesquisa documental, assim como para a pesquisa experimental, e divulgação do seu nome.

Durante a análise documental, coletaram-se informações do ambiente, dos especialistas nas situações investigadas, efetuando anotações a respeito da execução das atividades, das ações tomadas para correções e ajustes assim como dos resultados alcançados. Estes especialistas são os colaboradores designados pela empresa para a execução da atividade.

Os dados relativos à pesquisa documental foram referentes às atividades realizadas entre abril de 2013 e abril de 2014 (Fases 1 e 2) e entre janeiro de 2016 e abril de 2016 (Fases 3 e 4). A pesquisa experimental realizada na Fase 5 contempla o desenvolvimento do SE para apoio à classificação da criticidade da versão de *software*, que se encontra em andamento até o momento da elaboração deste artigo. Estas Fases serão demonstradas graficamente na Figura 3.

Durante a Fase 5 foi elaborado um questionário, utilizando-se indicadores provenientes das Fases 1 a 4 para ser aplicado aos especialistas na referida fase.

Com as estimativas apontadas anteriormente e de um breve estudo das bases de dados criadas pela execução do processo de gerenciamento de liberação de versão de *software* foi proposto então a elaboração de um questionário para a obtenção do conhecimento dos especialistas a ser aplicado aos mesmos que executam o referido processo para então alinhar as estimativas a uma real percepção dos especialistas aos valores aplicados aos KPIs. Em outras palavras absorver o conhecimento dos especialistas e através de um mecanismo de inferência inserir este conhecimento dentro de uma *shell* de desenvolvimento de Sistema Especialista.

As bases de dados utilizadas contemplam dois momentos da execução do processo de gerenciamento de liberação de versão de *software*. A primeira execução do processo sempre gera dados referentes a bases de homologação, onde a versão é implementada e classificada inicialmente. Posterior a este momento, quando a versão é direcionada para a produção, a execução do processo então gera dados referente as bases de produção.

O questionário foi elaborado através de reuniões com os especialistas responsáveis pela execução do processo de gerenciamento de liberação de versão de *software*, buscando-se obter o contexto da execução do processo, assim como também a pesquisa documental das Fases 1 a 4 ajudou neste entendimento.

4.2 Caracterização da Empresa

A empresa Softplan, foco do estudo, é uma desenvolvedora de *software*, situada na região sul do Brasil, a qual fornece suas soluções para clientes em todos estados brasileiros, América Latina e nos Estados Unidos. Atuante desde 1990 em vários segmentos de mercado: Justiça, Infraestrutura e Obras, Gestão Pública, Projetos Cofinanciados por Organismos Internacionais e Indústria da Construção; esta empresa possui um contingente de mais de 1500 colaboradores atuantes em suas sedes regionais e alocados dentro de seus clientes (Softplan, 2018).



A atenção será voltada as ao processo de gerenciamento de liberação de versão de *software* para as aplicações da área de automação judiciária do Primeiro Grau, entregue ao seu cliente de São Paulo, onde o número de usuários excede os 70.000. O estudo mostrou o problema que a equipe de infraestrutura de aplicação, alocada no cliente, encontrou no desenho, definição e execução do processo para o gerenciamento de liberação de versão de *software*, na busca de orientações focadas no BPM e ITIL e *Six Sigma*, assim como na classificação de criticidade de versão de *software* e nas soluções encontradas para que os objetivos da empresa junto ao cliente fossem alcançados.

A atividade *core* da área de infraestrutura de aplicação, desta empresa consiste na atualização do conjunto dos principais sistemas de automação do seu principal cliente. Quando o conjunto de aplicações foi fornecido a este cliente o processo de gerenciamento de liberação de versão de *software* era executado de forma não definida e manualmente por um único especialista, o qual detinha o conhecimento para si. Com a ampliação do ambiente de atualização, assim como a complexidade dos pacotes de liberação, surgiram erros na execução do processo que causaram problemas com o principal cliente da empresa.

4.3 Caracterização do Problema

Os problemas encontrados do processo de gerenciamento de liberação de versão de *software*, na empresa desenvolvedora, abordados neste artigo foram tratados em 5 Fases, conforme descritas a seguir:

- a) Fase 1: Implementação da metodologia BPM para o desenho e definição do processo de gerenciamento de liberação de versão de *software*, o qual não era ainda definido formalmente.
- b) Fase 2: Implementação das boas práticas da ITIL para a correta execução do processo de gerenciamento de liberação de versão de *software*, assim como a definição dos primeiros indicadores de execução e qualidade do processo.
- c) Fase 3: Definição do PMBOK como boas práticas de gestão de projetos para a correta execução da Fase 4, formalmente como um projeto interno da empresa.
- d) Fase 4: Implementação da metodologia *Six Sigma* para a melhoria da execução do processo de gerenciamento de liberação de versão de *software*, assim como melhoria nos indicadores anteriormente definidos na Fase 2.
- e) Fase 5: Obtenção do conhecimento dos especialistas e implementação de Sistema Especialista, junto a equipe de infraestrutura de aplicação da empresa desenvolvedora de *software* como o objeto de apoiar na classificação de criticidade de versão de *software*.

As 4 Fases iniciais compõem a pesquisa documental efetuada na empresa desenvolvedora de *software* que serviu como contexto para que a Fase 5 pudesse ser executada, compondo então a pesquisa experimental deste artigo. A Figura 3 demonstra graficamente o fluxo de execução das fases citadas.

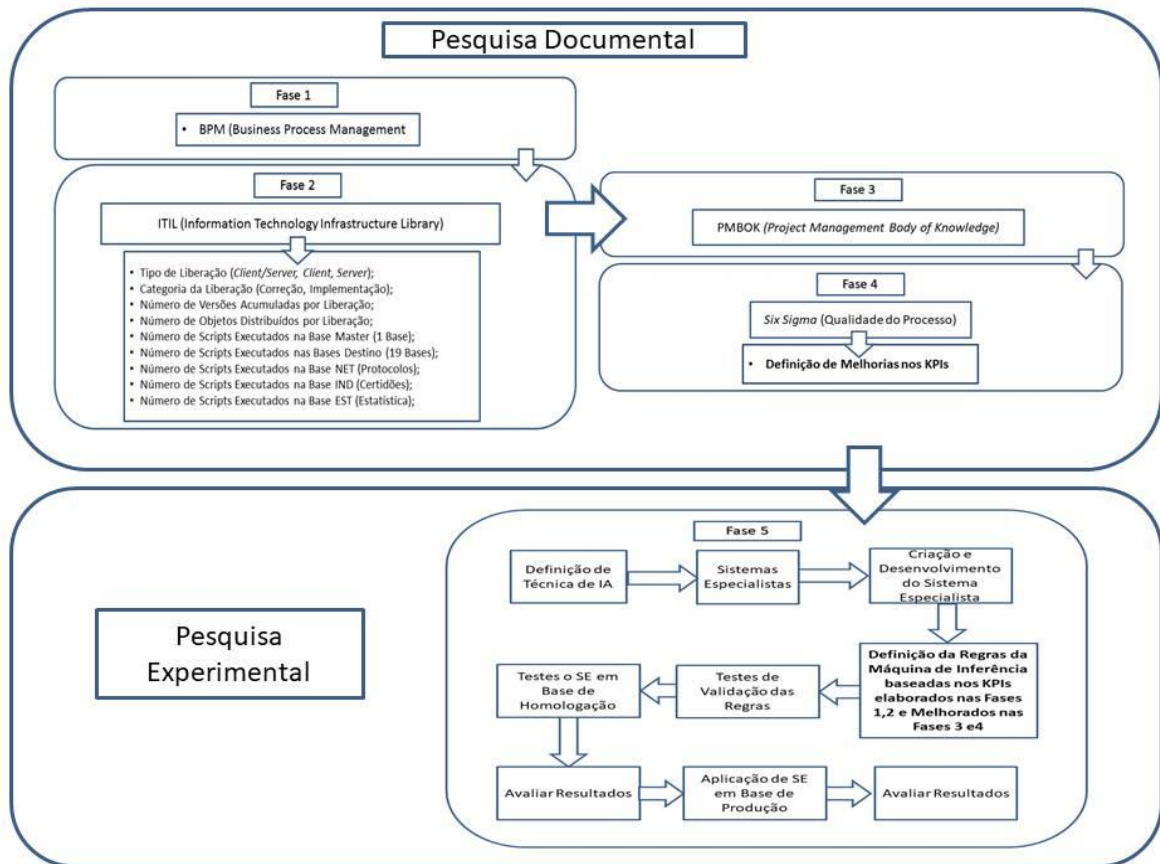


Figura 3: Fluxo de Fases da pesquisa
Fonte: Autores (2018)

A aplicação das metodologias e boas práticas (Fases 1 a 4) além de direcionarem a execução do processo de forma correta proporcionaram a criação de indicadores ou *Key-Performance Indicators* (KPIs) que favoreceram o desenvolvimento de uma base de dados com todos os fatores críticos da execução do processo, sendo alimentado sempre que o processo de gerenciamento de liberação de versão de *software* é executado.

Dentre os KPIs desenvolvidos na Fase 2 e melhorados na Fase 4 destacam-se os seguintes:

- Tipo de Liberação (*Client/Server, Client, Server*);
- Categoria da Liberação (Correção, Implementação);
- Número de Versões Acumuladas por Liberação;
- Número de Objetos Distribuídos por Liberação;
- Número de *Scripts* Executados na Base *Master* (1 Base);
- Número de *Scripts* Executados nas Bases Destino (19 Bases);
- Número de *Scripts* Executados na Base NET (Protocolos);
- Número de *Scripts* Executados na Base IND (Certidões);
- Número de *Scripts* Executados na Base EST (Estatística);

Estes KPIs são alimentados na base de dados pelos especialistas que executam o processo, onde a percepção da criticidade de cada liberação fica a critério de cada especialista que o executa. Esta percepção não é documentada formalmente o que mantém esta informação guardada pelo executor da atividade. Com o processo de gerenciamento de liberação de versão de *software* sendo executado continuamente pelos especialistas, esta



percepção de criticidade ficou limitada aos especialistas, direcionada pelo critério subjetivo de cada um. Hoje o processo é executado não mais por um, mas por cinco especialistas.

Neste contexto iniciou-se a Fase 5 com o objetivo de apoiar a classificação de cada versão de *software* em sua criticidade de modo formal e objetivo, onde os KPIs citados anteriormente sejam utilizados como balizadores para este entendimento e assim criando padrões desta percepção obtida pelos especialistas, deixando de ser subjetiva e pudesse se tornar objetiva, baseando-se nos KPIs já criados.

É proposta a criação de classificações tais como em relação a criticidade: versão Alta; versão Média e versão Baixa. Estas classificações poderão se tornar mais granulares se os KPIs também forem desmembrados em outros KPIs mais específicos. Porém o cenário atual reflete apenas os KPIs mencionados anteriormente. De acordo com uma análise informal dos indicadores, inicialmente uma versão de criticidade Alta pode receber esta classificação se os seguintes indicadores apresentarem valores elevados como indicado na Tabela 1.

Tabela 1: Estimativa de *Key-Performance Indicators* para versão Alta, Média e Baixa.

Fonte: Autores (2018)

<i>Key-Performance Indicators</i>	Valores Versão Alta	Valores Versão Média	Valores Versão Baixa
Número de Versões Acumuladas por Liberação	Acima de 5	Entre 2 e 4	1 Versão
Número de Objetos Distribuídos por Liberação	Acima de 40	Entre 17 e 39	Até 16
Número de <i>Scripts</i> Executados na Base <i>Master</i> (1 Base)	Acima de 10	Entre 3 e 9	Até 2
Número de <i>Scripts</i> Executados nas Bases Destino (19 Bases)	Acima de 190 por Base	Entre 20 e 189 por Base	Até 19

O critério para entender se os valores são elevados ou não, vai de acordo com o conhecimento que o especialista tem de acordo com a versão a ser analisada. Estas estimativas podem variar de especialista para especialista.

A ideia da aplicação de uma técnica de Sistema Especialista, é de conseguir absorver o conhecimento destes especialistas e através do mecanismo de inferência direcionar esta percepção criando formalmente uma classificação de criticidade para cada de versão de *software*, onde a percepção poderá ser formalmente divulgada para todas os interessados na execução do processo, esclarecendo a visão dos especialistas para todos os demais.

5 Apresentação e Discussão dos Resultados

A proposta de classificação da criticidade de versão de *software* baseou-se então em três categorias: Versão Alta, Média e Baixa. Para que os especialistas pudessem contemplar uma visão precisa de como esta classificação poderia ser realizada, chegou-se a um consenso que existiam seis cenários diferentes no processo de gerenciamento de liberação de versão de *software*. Cenários estes que eram reflexo do tipo, conteúdo e da categoria do pacote de liberação de versão de *software* que era disponibilizado pela equipe de desenvolvimento da empresa desenvolvedora de *software*, como demonstrado na Tabela 2.

Figura 2: Fluxo de ações de pesquisa durante a pesquisa

Fonte: Autores (2018)

Tipo e Categoria de Pacote de Liberação de Versão de <i>Software</i>	Propósito do Pacote de Liberação de Versão de <i>Software</i>
--	---



Server	Atualização apenas de Servidores de Aplicação
Client	Atualização apenas de Servidores de Borda / Estações de Usuários
Client/Server Correção	Atualização de Servidores de Aplicação/de Borda/Estações de Usuários) para correção de erros
Client/Server Implementação	Atualização de Servidores de Aplicação/de Borda/Estações de Usuários) para implementação de novas funcionalidades
PRECAT	Atualização de Servidores de Aplicação/de Borda/Estações de Usuários) apenas da base de dados PRECAT
EST	Atualização de Servidores de Aplicação/de Borda/Estações de Usuários) apenas da base de dados EST

O questionário então foi elaborado tendo como estrutura inicial o modelo representado na Tabela 3.

Tabela 3: modelo de cenário para aplicação de questionário para especialistas

Fonte: Autores (2018)

Cenário	Tipo de Liberação:		
	Categoria da Liberação:		
	Criticidade Baixa	Criticidade Média	Criticidade Alta
Número de Versões Acumuladas			
Número de bases PG5 atualizadas			
Número de Objetos PG5 Server			
Número de Objetos NET Server			
Número de Objetos IND Server			
Número de Objetos Extras			
Número de <i>Scripts</i> PG5 Master			
Número de <i>Scripts</i> PG5 Master/Destino			
Número de <i>Scripts</i> NET			
Número de <i>Scripts</i> IND			
Número de <i>Scripts</i> EST			

Para a primeira aplicação do questionário participaram do processo quatro analistas executores do processo e o coordenador de equipe. Dentre os analistas observou-se pontos de vista diferentes em relação as quantidades informadas em cada aspecto de classificação. Entendeu-se que critérios como tempo de equipe e experiência na execução do processo foram influenciadores pela diferença de opiniões.

O questionário, utilizando-se dos KPIs informados na seção 4.3 Caracterização do Problema buscou consolidar este conhecimento abordando cenários do processo de liberação de versão de *software* tendo como diferenciadores o seu conteúdo, tipo e a categoria dos pacotes de liberação de versão.

Após a coleta e análise dos resultados dos questionários, uma nova reunião foi realizada para então chegar-se a um consenso da criticidade de cada cenário de liberação de versão de *software*, e consolidação dos dados. Com os dados consolidados então obteve-se um modelo de criticidade de versão de *software* para cada cenário anteriormente estipulado como demonstrado no exemplo da Tabela 4.

Tabela 4: Recorte do modelo de criticidade para o cenário de liberação de versão de *software*

Fonte: Autores (2018)

1º Cenário	Tipo de Liberação: Server
------------	---------------------------



	Categoria da Liberação: Correção		
	Criticidade Baixa	Criticidade Média	Criticidade Alta
Número de Versões Acumuladas	1 até 3	4 até 6	7 ou mais
Número de Bases PG5 Atualizadas	1	2 até 23	24 ou mais
Número de Objetos PG5 Server	1 até 3	4 ou 5	6 ou mais
Número de Objetos NET Server	1 até 3	4 ou 5	6 ou mais

Como exemplo de *shell* de desenvolvimento de SE temos o Expert SINTA. O Expert SINTA é um conjunto de ferramentas computacionais fundamentadas em técnicas de IA para geração automática de SEs. Criados por alunos do grupo SINTA (Sistemas INTeligentes Aplicados) da Universidade Federal do Ceará. Este conjunto de ferramentas foi desenvolvido com o intuito de simplificar a implementação de SE, aliviando o projetista da tarefa de programação e criação de máquinas de inferência, bem como a implementação da interface com o usuário. Além disso, o Expert SINTA dispõem de ferramentas de depuração e explicação de resultados atingidos durante uma consulta, gerando uma árvore que representará todos os passos efetuados até a conclusão de determinado objetivo (Lia, 2017).

Iniciou-se então uma análise para a criação do primeiro protótipo do SE, sendo necessário estipular todas as variáveis que seriam utilizadas, assim como suas categorias (Numérica, Univalorada ou Multivalorada), bem como atribuir valor a elas.

A definição da variável-objetivo também foi necessária neste momento onde a variável Criticidade da Versão seria então definida como Alta, Média ou Baixa, ao término da inferência pelo SE. Por conseguinte, teve início a definição da interface com o usuário uma vez que sua elaboração torna a interação entre o usuário e o SE mais fácil de se compreender. Por padrão o Expert SINTA cria uma interface genérica o que torna a experiência inicial pouco esclarecedora sobre o que o sistema questiona ao usuário, então torna-se um dos fatores fundamentais a elaboração da interface com o usuário, sendo este um dos elementos que compõe o SE de acordo com o referencial teórico.

Como próximo passo iniciou-se a elaboração das regras de produção que servem como componentes da base de conhecimento a ser inserida no SE. Estas regras de produção são exatamente um reflexo das informações consolidadas pelos especialistas nos questionários aplicados anteriormente. Como foram definidos seis cenários específicos para a execução do processo de gerenciamento de liberação de versão de *software*, iniciou-se a criação de um conjunto de regras de produção para cada cenário, onde o SE deve executar de forma isolada a inferência para cada cenário.

A seguir é demonstrado um exemplo de regra de produção aplicado no Expert SINTA para a montagem da base de conhecimento.

SE

Tipo de Liberação = Server

E 1_Número de Versões Acumuladas = 1 até 3

E 1_Número de Bases PG5 Atualizadas = 1

E 1_Número de Objetos PG5 Server = 1 até 3

E 1_Número de Objetos NET Server = 1 até 3

E 1_Número de Objetos IND Server = 1 até 3

E 1_Número de Objetos Extras = 1

E 1_Número de Scripts PG5 Master = 1 até 10

E 1_Número de Scripts PG5 Master/Destino = 1 até 10

E 1_Número de Scripts NET = 1 até 5

E 1_Número de Scripts IND = 1 até 5

E 1_Número de Scripts EST = 1

ENTÃO



Criticidade de Versão = Criticidade Baixa CNF 100%

Esta regra foi gerada através das informações coletadas dos questionários aplicados aos especialistas, e discutidas em reunião, chegando-se a um consenso dos intervalos que corresponderiam a uma versão de Criticidade Baixa, de acordo com o Tipo de pacote de liberação de versão de *software*, no caso Tipo de Liberação = Server.

O critério de criação das regras de produção exigiu uma análise de como cada regra deveria ser inserida de forma a isolar os cenários, pois apesar de algumas variáveis terem o mesmo significado em cada cenário, os valores assumidos em cada um poderiam se sobrepor se fossem atribuídos dentro de uma variável comum a todos cenários. Então definiu-se um conjunto de regras específicas para cada cenário, assim como seus valores, oriundos do questionário consolidado em reunião com os especialistas.

Inicialmente as regras foram criadas de forma a obedecer a uma sequência ideal de como os valores dos indicadores foram apresentados, ou seja, se todas as respostas aos questionamentos do SE apontassem para uma sequência ideal de respostas, a classificação da criticidade seria a já esperada.

Durante os primeiros testes foi utilizada como referência a base de dados utilizada em homologação. Verificou-se que as regras de produção inseridas no SE atenderam aos critérios de criticidade apresentados e coletados nos questionários aplicados.

No decorrer dos testes os especialistas apontaram que apesar dos pacotes de liberação de versão terem um padrão de envio pela equipe de desenvolvimento dentro de cada cenário, possíveis variações poderiam ocorrer em relação a número de objetos e *scripts* encaminhados, assim como o número de versões acumuladas por liberação.

Este posicionamento por parte dos especialistas demonstrou a necessidade de um refinamento da elaboração das regras da base de conhecimento. Este refinamento proporciona que possíveis variações dos valores das variáveis, dentro de cada cenário, permitam que o SE se mantenha dentro da sequência de questionamentos e assim inferir o resultado de classificação de criticidade esperado, contemplando cada possível variação.

Estas possíveis variações estão sendo tratadas pelos especialistas para gerar possíveis novas regras ou o ajuste das regras de produção já existentes. É previsto novas reuniões com os especialistas para o entendimento e formulação destas possíveis variações.

A interface amigável facilitou o entendimento e a aceitação do SE pelos especialistas que se mostraram muito colaborativos com os testes de validação do primeiro protótipo utilizando a base de dados de homologação.

6 Conclusão

O Sistema Especialista desenvolvido e aplicado na execução do processo de gerenciamento de liberação de versão de *software* para apoiar o especialista na classificação da criticidade da versão de *software* tem sido validado pelos próprios especialistas no decorrer dos testes com a base de dados de homologação.

Foi verificado que em alguns pacotes de liberação o número de objetos permanecia o esperado, porém o número de *scripts* sofria alteração devido a alguma necessidade solicitada no desenvolvimento do *software*. Essa alteração saía do padrão esperado pelo tipo do pacote, consequentemente alterando a categoria da liberação naquele cenário específico, como demonstrado na Tabela 3.

Esta variação mostrou a necessidade de uma análise mais profunda das regras de negócio, buscando-se abranger estas possíveis variações que podem surgir de forma aleatória no pacote de liberação de versão de *software*.



Então consegue-se inferir através das respostas aos questionamentos do SE que uma versão tem criticidade Alta, Média ou Baixa, quando os pacotes de liberação são encaminhados pela equipe de desenvolvimento seguindo o padrão previsto pelos especialistas.

Até o momento do desenvolvimento deste artigo o Sistema Especialista está sendo testado e depurado pelos especialistas com bases de dados de homologação. Como estudos futuros, após a validação do SE pelos especialistas, utilizando a base de dados de homologação, será utilizada então a base de produção que possui um conjunto maior de dados, e será avaliada também a inserção de novas variáveis que são relevantes no ambiente de produção de liberação de *software* como número de analistas envolvidos no processo, assim como tempo de execução da liberação de versão de *software*.

7 Referências Bibliográficas

- ABPMP, (2018). Association of Business Process Management Professionals. Guia para o Gerenciamento de Processos de Negócio Corpo Comum de Conhecimento. Disponível em < <http://www.abmp-br.org/sobre-a-abmp-brasil/>>. Acesso em: 10 de julho de 2018.
- Adeli, H. (2014). *Expert systems in construction and structural engineering*. CRC Press.
- Axelos, (2013). Global Best Practice. *ITIL Maturity Model and Self-Assessment Service: User Guide*. Axelos Limited. Disponível em <http://www.axelos.com>. Acessado em 10 de julho de 2018.
- Barros, M. D., Salles, C. A. L., Gomes, C. F. S., da Silva, R. A., & Costa, H. G. (2015). Mapping of the Scientific Production on the ITIL Application Published in the National and International Literature. *Procedia Computer Science*, 55, 102-111. DOI 10.1016/j.procs.2015.07.013
- Calazans, A. T. S., Kosloski, R. A. D., & de Albuquerque Guimarães, F. (2016). Proposta de modelo de medições para contratação do gerenciamento de processo de negócio (Business Process Management-BPM). *JISTEM-Journal of Information Systems and Technology Management (Online)*, 13(2), 275-300. DOI: 10.4301/S1807-17752016000200007.
- Estorílio, C. C. A., & Amitrano, F. G. (2013). Aplicação de Seis Sigma em uma empresa de pequeno porte. *Produto & Produção*, 14(2), 1-25. DOI: <https://doi.org/10.22456/1983-8026.20505>
- Erdogan, A., & Canatan, H. (2015). Literature Search Consisting of the Areas of Six Sigma's Usage. *Procedia-Social and Behavioral Sciences*, 195, 695-704.
- Farias, E. B. P., & Sassi, R. J. (2018). Framework ITIL e Inteligência Computacional na padronização do atendimento do Service Desk de um Hospital Público. *Revista Eletrônica Gestão & Saúde*, 9(2), 219-233. DOI: <http://dx.doi.org/10.18673/gv9i2.24246>
- Fernandes, A. A., & De Abreu, V. F. (2014). *Implantando a Governança de TI-: Da estratégia à Gestão de Processos e Serviços*. Brasport.
- Frizon, M., Silveira, S. R. & Cunha, G. B. (2015). Desenvolvimento de um Protótipo de Sistema Inteligente de Apoio à Decisão como Ferramenta de Business Intelligence: um estudo de caso na Arbaza Alimentos. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica, Franca*, 6(5),16-36.
- Gatto, D. D. O., Sassi, R. J., & Costa, I. (2017). Metodologia BPM e Framework ITIL no Processo De Gerenciamento De Liberação De Versão em Empresa Desenvolvedora De Software In: *XIV SEGeT – Simpósio em Excelência e Gestão de TI, Resende - RJ*.
- Gatto, D. D. O., Farias, E. B. P., & Sassi, R. J.. (2017). Metodologia Six Sigma com Base no Guia PMBOK no Processo de Gerenciamento de Liberação de Versão em Empresa Desenvolvedora de Software In: *VI SINGEP – Simpósio Internacional de Gestão de Projetos, Inovação e sustentabilidade, São Paulo - SP*.



- Gupta, S; Singhal, R. (2013) Fundamentals and characteristics of an expert system. *International Journal on Recent and Innovation Trends in Computing and Communication*,. 1 (3), 110-113.
- ITIL Service Lifecycle Publication Suite, Editora TSO; Edição: UK ed., 2013.
- Koche, J. C. (2003). Fundamentos de Metodologia Científica. Petrópolis: Vozes.
- LIA. ExSinta versão 1.1 (2018). Uma ferramenta visual para criação de Sistemas Especialistas manual do usuário. Laboratório de Inteligência Artificial. Disponível em: <<http://www.lia.ufc.br>>. Acesso em: 10 de junho 2018.
- Liao, S. (2018). Expert system methodologies and applications – a decade review from 1995 to 2004. *Expert Systems with Applications*. v. 28, p. 93-103, 2005. Disponível em <<http://www.periodicos.capes.gov.br>>. Acesso em: 10 de julho de 2018.
- PMI, (2018). Um Guia do Conhecimento em Gerenciamento de Projetos. Guia PMBOK 6ª Edição - EUA: Project Management Institute.
- Six Sigma Institute. (2018) “What Is Six Sigma?”, disponível em <www.sixsigma-institute.org/Six_Sigma_Roles_And_Responsibilities.php>. Acesso em: 10 de junho de 2018.
- Softplan. (2018). Santa Catarina. Disponível em: <<http://www.softplan.com.br/a-softplan/quem-somos/>> Acesso em: 10 de julho de 2018.
- Vitoriano, M. A. V., & Souza Neto, J. (2015). Information technology service management processes maturity in the Brazilian Federal direct administration. *JISTEM-Journal of Information Systems and Technology Management*, 12(3), 663-686. - DOI: 10.4301/S1807-17752015000300009.
- Wagner, W. P. (2017). Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies. *Expert Systems with Applications*, 76, 85-96. DOI: 10.1016/j.eswa.2017.01.028