

## **INFLUÊNCIA DAS PRÁTICAS DO DEVOPS NOS PROCESSOS DE GESTÃO DE TI CONFORME O MODELO COBIT 5**

**TEREZA CRISTINA MAIA FERNANDES**  
INSTITUTO DE TECNOLOGIA ARAGON & COSTA  
crismaia@itac.org.br

**IVANIR COSTA**  
UNIVERSIDADE NOVE DE JULHO (UNINOVE)  
icosta11@live.com

**NILSON SALVETTI**  
UNIVERSIDADE NOVE DE JULHO (UNINOVE)  
nilson.salvetti@hotmail.com

**FÁBIO LUÍS FALCHI DE MAGALHÃES**  
UNIVERSIDADE NOVE DE JULHO (UNINOVE)  
fabiosimp@gmail.com

**AGUINALDO ARAGON FERNANDES**  
INSTITUTO DE TECNOLOGIA ARAGON & COSTA  
aragon@aragonconsultores.com.br

# INFLUÊNCIA DAS PRÁTICAS DO DEVOPS NOS PROCESSOS DE GESTÃO DE TI CONFORME O MODELO COBIT 5

## 1. INTRODUÇÃO

Com o avanço de novas tecnologias como a computação em nuvem, *'big data'*, *'analytics'*, mídias sociais, *'mobile'*, inteligência cognitiva e internet das coisas, ou que Ross *et al.* (2016) denomina de SMACIT (*Social, Mobile, Analytics, cloud and IOT*), as quais estão dando sustentação para a transformação digital das organizações, observa-se uma mudança significativa no modo como as empresas desenvolvem produtos e serviços, aprimoram a experiência dos seus clientes, mudam a forma de operar seus processos e também propiciam o desenvolvimento de novos modelos de negócio.

Neste cenário, o software, como componente lógico de uma solução de tecnologia de informação, tem se tornado parte fundamental dessa transformação, que chega numa velocidade nunca antes experimentada pelas organizações e que está exigindo respostas rápidas também por parte da TI (Taurion, 2016).

Uma das respostas às necessidades de implementações rápidas e contínuas foi a verdadeira explosão no uso dos métodos ágeis, com a redução significativa dos prazos de desenvolvimento do software. Entretanto, a instalação e operação dos sistemas ou *Deployment*, como será utilizado neste texto, não avançou na mesma velocidade (Sato, 2013).

Empresas tradicionais levam meses para que uma demanda chegue à produção. Porém, com o ciclo menor de vida útil de produtos e serviços, obriga as organizações se tornarem mais ágeis e enxutas, em que a demanda solicitada pelo negócio demore no máximo horas ou dias para estar disponível na produção. Exemplos são empresas inovadoras, como o Facebook e Amazon (Freire, 2013).

Outra questão essencial para o sucesso do negócio utilizando uma infraestrutura ágil é a integração entre o desenvolvimento e a operação. A computação em nuvem e a virtualização são outras duas tendências dentro deste movimento em relação a infraestrutura ágil (Taurion, 2014).

De acordo com Rappaport (2014), para a resolução desse hiato entre o desenvolvimento e a infraestrutura, a comunidade de TI deu início ao movimento DevOps. Para Kim (2013), o termo DevOps refere-se ao movimento profissional emergente que advoga uma colaboração entre as equipes de desenvolvimento e operações de TI resultando em um fluxo rápido de trabalho planejado, enquanto, simultaneamente, aumenta a confiabilidade, estabilidade e resiliência do ambiente de produção.

A ISACA (2015) lançou um documento denominado de *'DevOps Practitioner Considerations'* onde demonstra a influência dessas práticas na governança da tecnologia da Informação. Neste estudo são recomendados controles chaves para que as empresas que desejam adotar o *DevOps* devam considerar para reduzir custos e obter maior agilidade. Isso engloba, por exemplo, o gerenciamento de dependências, treinamento dos desenvolvedores em segurança, *logs* de acesso e atividades, gerenciamento do desempenho da aplicação, dentre outros. Por outro lado, de acordo com Fernandes e Abreu (2014) as tecnologias emergentes têm impacto significativo nos processos de governança e gestão de TI.

Diante do exposto, surge a seguinte pergunta que norteia esta pesquisa: **Qual é a influência que as práticas do DevOps têm sobre os processos de gestão de TI conforme o modelo de referência do COBIT em sua versão 5?**

Assim, o objetivo desta pesquisa é **explorar a influência que as práticas do DevOps têm sobre os processos de gestão de TI baseados no padrão COBIT 5 de Governança de TI, visando identificar os processos mais afetados, subsidiando assim iniciativas de implantação de processos de TI com base neste *framework* da ISACA (2012b).**

Em uma organização que deseja seguir o *framework* COBIT para fins de governança, *compliance* e aprimoramento do alinhamento da TI ao negócio, assim como o aprimoramento da entrega dos serviços de TI e que deseja implantar práticas DevOps, o entendimento do impacto dessas práticas sobre as práticas de gestão do COBIT é de fundamental importância, de forma a guiar a instânciação do processo.

Este trabalho foi organizado como segue: após uma breve introdução, na seção 2 é apresentada uma revisão da literatura sobre o DevOps e o COBIT 5. Na seção 3, a metodologia de pesquisa aplicada no trabalho é descrita. Na seção 4, é exibida a análise dos resultados e finalmente, na última e seção 5, expõem-se as considerações finais da pesquisa.

## 2. FUNDAMENTAÇÃO TEÓRICA

Neste item serão apresentados os conceitos e definições necessárias para a fundamentação teórica visando o entendimento proposto pelo trabalho de pesquisa.

### 2.1 DevOps

O objetivo do movimento DevOps é fomentar uma cultura de colaboração entre as equipes de desenvolvimento e de operações, que através de um fluxo bidirecional de comunicação contínuo e de compartilhamento não só de resultados, mas também de ideias, o que permite tornar a TI mais ágil e controlada, melhorando as entregas das necessidades de negócios com velocidade e confiabilidade (Erich, Amrit & Maya, 2014).

Para Wettinger, Andrikopoulos & Leymann (2015), o DevOps é um paradigma emergente que tem por objetivo eliminar as barreiras entre o pessoal de desenvolvimento e de operações.

Zentgraf (2013) preconiza que uma organização necessita entregar funcionalidades de software a um ritmo constante, contínuo e de forma sustentável. O autor também propõe uma taxonomia de recursos para o DevOps composto por elementos como a gestão da mudança, a orquestração, o *Deployment* da aplicação, o monitoramento da aplicação em ambiente de produção e o fornecimento da configuração apropriada da infraestrutura tecnológica.

Erich *et al.* (2014) realizando uma revisão da literatura sobre DevOps encontrou os seguintes rótulos quando a literatura aborda o tema: cultura de colaboração, automação, medição, compartilhamento de documentação e conhecimento, serviços, garantia da qualidade e estruturas/padrões.

Wahaballa, Wahaballa, Abdellatif, Xiog & Quin (2015) propõem um modelo unificado para o DevOps considerando neste modelo elementos como um modelo de aplicação e de dados, modelos de fluxo de trabalho para o processo de software até a ‘Entrega Contínua’ e um modelo de infraestrutura que abrange desde a infraestrutura básica até a infraestrutura com ‘*scripts*’ automáticos de instalações.

### 2.2 Pilares do Devops

Conforme Sharma (2014), o DevOps se mantém atualmente em cinco pilares principais, conhecidos pelas siglas CALMS: *Culture* (Cultura); *Automation* (Automação); *Lean* (Enxuto); *Measurement* (Medição); *Sharing* (Compartilhamento).

DevOps é uma cultura de colaboração, compartilhamento, de entrega, calcada em ferramentas de apoio e na automação de atividades que altera a cultura organizacional para a criação e utilização de processos ágeis, que aproxima o desenvolvimento de software da operação de TI reduzindo o gap entre as duas áreas, levando ao sucesso de entregas rápidas, aumento de qualidade e alinhamento com as demandas de negócio (Braga, 2015).

Segundo Reddy (2013), centrado na preservação do valor, os princípios *lean* e ágeis se aplicados em todo o ciclo de vida da entrega de softwares do DevOps garantem a entrega rápida de produtos de alta qualidade, reduz o desperdício e custos, aumentando a competitividade das empresas através de inovação e aprendizado contínuo fornecendo as organizações o aumento das oportunidades de mercado e a redução no tempo do *feedback* do cliente.

As medições fornecem as evidências de modo que todos possam gerar as oportunidades de melhoria, monitoram processos e condições em tempo real, identificam gargalos em um sistema de aplicativo ou gerenciam requisitos de capacidade proativa e com essas informações, a equipe pode determinar que passos seguir (Riley, 2015).

O compartilhamento de conhecimentos entre as equipes no Devops é essencial, pois ter uma boa comunicação entre as equipes e criam um excelente canal de *feedback* que fomentam um processo de melhoria contínua (Huttermann, 2012; Kim, 2013).

## 2.3 Práticas DevOps

Uma questão decisiva que pode influenciar tanto para o sucesso como também o fracasso de uma iniciativa em uma organização é o tempo investido entre o desenvolvimento de um recurso e o *deploy* em produção. O DevOps se utiliza de um conjunto de práticas dentro da cultura ágil que permitem que as mudanças sejam levadas rapidamente para produção de forma coordenada e com qualidade, reduzindo o *time to market* (Humble, 2010).

Segundo a literatura apresentada nesse trabalho as práticas ágeis mais utilizadas pelo DevOps são discutidas a seguir.

### 2.3.1 Integração Contínua

É uma prática de desenvolvimento de software que teve origem no eXtreme Programming (XP) e consiste que o desenvolvedor integre continuamente o código alterado e/ou desenvolvido ao projeto principal, permitindo assim detectar, além de localizar, erros rapidamente - podendo haver múltiplas integrações por dia (Fowler, 2006).

A integração contínua pode trazer diversos benefícios para a organização e dentre eles pode-se citar um tempo menor de depuração, maior adição de características do software, redução de problemas e um menor tempo de integração, bem como o aumento de visibilidade e comunicação entre as equipes (Duvall, Matyas & Glover, 2007).

### 2.3.2 Entrega Contínua

Entrega contínua é uma prática que garante a entrega de software da equipe de desenvolvedores para o ambiente de produção em um processo confiável, previsível, visível e o mais automatizado possível, com riscos quantificáveis e bem entendidos (Humble & Farley, 2013). Ao invés de planejar grandes *releases*, a TI deve elaborar software em ciclos mais curtos, garantindo que o novo código possa ser implantado no ambiente de produção a qualquer momento de forma eficiente, sem comprometer a qualidade (Duvall, 2011).

### 2.3.3 Deployment Contínuo

*Deployment* Contínuo é uma prática de engenharia de software, que começa onde a Integração Contínua (IC) termina. É a ação de instalar um pacote do *software* de forma automática e sistêmica, ou seja, toda vez que o *software* passar por todas as fases da Integração Contínua (baixar o código, integrar, *build*, rodar os testes e gerar o artefato) e se criar o pacote em ‘estado de pronto’ é disparado o processo de *Deployment* e o *software* é instalado em um determinado servidor (Humble & Farley, 2013).

### 2.3.4 Monitoramento Contínuo

O Monitoramento Contínuo é uma prática que ajuda os times de desenvolvimento e operações a identificar rapidamente quando um serviço está indisponível, entender as causas subjacentes e, sobretudo, aplicar esses aprendizados para antecipar os problemas antes que ocorram. Além disso, determinam predisposição para utilização, com a vantagem do desenvolvimento iterativo, compreendendo o mais correto o comportamento do aplicativo com o tempo (Palko, 2015).

### 2.4 Infraestrutura como Código (IAC)

A infraestrutura como código (IAC) é um tipo de infraestrutura de TI, na qual os times de infraestrutura e operação podem gerenciar configurações e automatizar o provisionamento da infraestrutura, além de implementações e realizar o fornecimento de serviços através de código escrito. Isso irá eliminar um processo manual seja para a configuração tanto para os servidores ou serviços, ou seja, gerenciar a infraestrutura como um sistema de software, com funcionalidades bem testadas, tarefas operacionais e rotineiras de gerenciamento, atualização e documentação da infraestrutura de forma segura e em larga escala (Humble & Farley, 2013).

### 2.5 COBIT

O COBIT foi desenvolvido pela ISACA e, atualmente em sua versão 5, é formado pelo modelo em si, com cinco princípios e sete habilitadores. Esse modelo abarca um modelo de referência de processo que descreve em detalhes uma série de processos de governança e de gestão relacionados às ações de TI (ISACA, 2012b; Magalhães, Gaspar, Costa, & Campos, 2017). Ademais, este é um dos modelos mais comuns utilizados em empresas brasileiras (Lunardi, Dolci, Maçada, & Becker, 2014).

Enquanto o objetivo de um *framework* é gerar maior valor competitivo para o negócio, o COBIT especificamente se preocupa que a TI seja gerida de forma holística e integrada, possibilite a realização de benefícios, a melhoria dos níveis de risco e a otimização do uso dos recursos de TI (ISACA, 2012b; Selig, 2016).

Giampaoli, Testa & Luciano (2011) consideram que a utilização deste *framework* deve ser moldada de acordo com a cultura organizacional, porém sua aplicação não assegura que a empresa terá o resultado esperado. O mesmo pode ser utilizado tanto para apoio na estratégia como para a operação da TI (Fernandes & Abreu, 2014).

O manual do *framework* COBIT 5 (ISACA, 2012a) define um domínio específico para atender aos processos governança de TI, denominado de ‘Avaliar, Dirigir e Monitorar’, do inglês, *Evaluate, Direct and Monitor* (EDM). Ademais, quatro outros domínios são responsáveis pelas questões de planejamento, construção, execução e monitoramento, denominados, respectivamente: ‘Alinhar, Planejar e Organizar’, do inglês, *Align, Plan and Organise* (APO); ‘Construir, Adquirir e Implementar’, do inglês, *Build, Acquire and Implement* (BAI); ‘Entregar, Serviços e Suporte’, do inglês, *Deliver, Service and Support* (DSS); ‘Monitorar, Avaliar e Analisar’, do inglês, *Monitor, Evaluate and Assess* (MEA).

O COBIT foi selecionado como *framework* para comparação com as práticas DevOps por ser considerado o modelo mais amplo e voltado à governança e gestão de TI de maneira mais abrangente (ISACA, 2012b).

### 3. MÉTODO DE PESQUISA

A partir do objetivo dessa pesquisa, que é o de explorar a influência das práticas do DevOps têm sobre processos de gestão de TI baseados no padrão COBIT 5 foi definido o método de natureza qualitativa, através de pesquisa bibliográfica em periódicos e revistas profissionais, pesquisa documental e utilização do instrumento teste de face, isto é, através da validação por especialistas (Audy & Prikladnicki, 2007).

Conforme Giampaoli *et al.*, (2011) os processos dos domínios de Adquirir e Implantar e Entregar e Suportar são os mais propensos em serem observados pelos CIOs (*Chief Information Officers*), pois são os que maior influência têm sobre a parte operacional. Dessa forma, este estudo se circunscreve nos processos dispostos nos domínios BAI e DSS, ou ‘Construir, Adquirir e Implantar’ e ‘Entregar, Serviço e Suporte’, respectivamente, do *framework* COBIT em sua versão 5.

O critério de seleção dos processos é baseado na aderência de cada prática DevOps às práticas considerando uma abordagem qualitativa do ponto de vista dos autores, conforme está demonstrado no Quadro 1.

Considerando o exposto na fundamentação teórica, podem-se elencar como práticas fundamentais do DevOps, a Integração Contínua, a Entrega Contínua, a Infraestrutura como Código e o Monitoramento Contínuo que são as utilizadas neste estudo.

**Quadro 1- Análise de aderência das práticas DevOps aos processos COBIT dos domínios selecionados**

Processo COBIT	Nome do Processo	Práticas DevOps				
		Integração contínua	Entrega contínua	Deployment contínuo	Infraestrutura como código	Monitoramento contínuo
BAI01	Gestão de programas e projetos	O ciclo DevOps pode ser elemento necessário para a gestão do desenvolvimento do software. Entretanto do ponto de vista operacional o processo BAI01 praticamente não tem relação com as práticas DevOps referenciadas.				
BAI02	Gerenciar definição de requisitos	O mesmo raciocínio acima se aplica ao BAI02. No DevOps parte-se do pressuposto de que os requisitos já foram definidos. Portanto, a relação entre este processo e as práticas DevOps é inexistente.				
BAI03	Gerenciar identificação e desenvolvimento de soluções	Este processo COBIT trata do desenvolvimento em si da solução, no caso aqui do software abrangendo também o processo de teste. Analisando o conceito de cada uma das práticas DevOps a Integração e Entrega Contínua e a Infraestrutura como Código parecem ter forte relação com o processo.				
BAI04	Gerenciar disponibilidade e capacidade	Este processo COBIT foca no gerenciamento da disponibilidade e capacidade de curto e longo prazo. Para o DevOps é de especial interesse o monitoramento da disponibilidade e capacidade para atender à operação do software. As práticas DevOps que parecem ser mais relacionadas são a Infraestrutura como Código e o Monitoramento Contínuo.				
BAI05	Gerenciar capacidade de mudança organizacional	Este processo COBIT foca exclusivamente no planejamento e implantação de mudanças organizacionais não tendo, nesse sentido, relação com as práticas DevOps.				
BAI06	Gerenciar mudanças	Este processo COBIT foca no gerenciamento de mudanças na configuração tanto de software como de infraestrutura. O gerenciamento de mudança, apesar de ter uma conotação administrativa, se relaciona com todo o ciclo do DevOps da Integração Contínua ao Deployment Contínuo e a Infraestrutura como Código.				
BAI07	Gerenciar aceitação e transição da mudança	Este processo COBIT foca nos testes de aceitação e na promoção do software para a produção. Assim sendo, parece ter forte relação com a Entrega Contínua e o Deployment Contínuo.				
BAI08	Gerenciar conhecimento	Um dos pilares do DevOps é o compartilhamento do conhecimento. Operacionalmente, em um <i>pipeline</i> automatizado, a gestão da configuração e os repositórios de artefatos devem estar disponíveis para todos os envolvidos no processo. Dessa forma este processo parece ter forte relação com todas as práticas do ciclo do DevOps.				
BAI09	Gerenciar ativos	Este processo COBIT foca no gerenciamento de ativos, hardware, software, licenças de aplicativos considerando desde a contratação, internalização até a renovação e descarte. O gerenciamento dos ativos é crítico para a entrega da configuração necessária para o processamento do ciclo DevOps. Tem, portanto, relação com as práticas DevOps.				
BAI10	Gerenciar a configuração	Este processo do COBIT foca no gerenciamento de toda a configuração de hardware e software disponibilizada para o ciclo DevOps. De acordo com a Humble (2010) a gestão da configuração é crítica para o ciclo DevOps. Dentro desta ótica, este processo parece estar fortemente relacionado com todas as práticas DevOps.				
DSS01	Gerenciar operações	Este processo COBIT foca no gerenciamento das instalações, na aplicação de procedimentos operacionais de produção, no monitoramento da infraestrutura de TI e no gerenciamento dos fatores ambientais que possam afetar equipamentos. Parece ter relação com o <i>Continuous Deployment</i> , a Infraestrutura como Código e o Monitoramento Contínuo.				
DSS02	Gerenciar solicitações e incidentes de serviços	Este processo COBIT foca no atendimento a solicitações de serviços e ao gerenciamento da resolução de incidentes. No ciclo DevOps, a resolução de falhas operacionais de funcionalidades no software deve ser feita imediatamente a partir dos eventos que estão sendo monitorados. Neste processo o registro de incidentes é registrado a posteriori. Por conseguinte, parece ter pouca relação com as práticas DevOps.				
DSS03	Gerenciar problemas	Este processo COBIT foca na identificação e proposição de solução para incidentes que se repetem. Não se identifica relação com as práticas DevOps.				
DSS04	Gerenciar continuidade	Este processo COBIT foca no planejamento da continuidade dos serviços de TI. Sua relação com as práticas DevOps não foi percebida.				
DSS05	Gerenciar serviços de segurança	Este processo COBIT foca nos aspectos operacionais da segurança da informação no ambiente de TI, principalmente no tocante a segurança lógica e física. Sua relação com as práticas DevOps não foi percebida a não ser com a incorporação ou integração com softwares de segurança existentes no ambiente de operações.				
DSS06	Gerenciar controles do processo de negócio	Este processo COBIT foca nos processos requeridos para manter a integridade de informações que são processadas para os processos de negócio. Também não parece ter relação com as práticas DevOps.				

Fonte: Os autores

Conforme a análise realizada pelos autores, os processos COBIT a serem utilizados pelo estudo para avaliar a influência das práticas DevOps são oito, apresentados no quadro 1 acima.

Em seguida, criou-se uma ferramenta chamada Matriz de Influência das práticas DevOps nos Processos de TI (MIDPTI), conforme extrato exemplo apresentado na Tabela 1, que gera uma relação cruzada entre as práticas DevOps e os processos de TI. No exemplo abaixo, está sendo apresentadas apenas as práticas de gestão específicas do processo BAI06 - gerenciar mudanças do COBIT 5 e as cinco práticas selecionadas do DevOps.

**Tabela 1 - Modelo da matriz da influência das práticas DevOps nos processos de TI (Exemplo)**

Práticas DevOps	Processo COBIT - BAI06 - Gerenciar Mudanças				
	Processos Cobit	BAI06.01	BAI06.02	BAI06.03	BAI06.04
	Grau de Influência	1-5	1-5	1-5	1-5
Entrega contínua		1	4	3	4
Implantação Contínua		2	2	4	5
Integração Contínua		3	3	2	4
Infraestrutura como código		2	2	1	5
Monitoramento Contínuo		3	3	5	5

Fonte: Os autores

Após a construção da (MIDPTI), foi feito um teste de face com três especialistas em COBIT, todos com certificações COBIT *Foundation* e *Certified in the Governance Enterprise IT* e ITIL Expert com grande experiência profissional na área de gerenciamento de operações e de serviços de TI de uma forma geral.

Apesar da existência de uma escala de avaliação proposta pelo COBIT (ISACA, 2012a), foi proposta pelos especialistas consultados, a utilização de uma pontuação da escala Likert para avaliar a influência das práticas *DevOps*, conforme Tabela 2. Esta análise foi feita baseada nas práticas de gestão de cada processo COBIT, mensurando o percentual de práticas influenciadas.

**Tabela 2 - Escala Likert de influência nos processos**

Nível	Grau de influência	Métrica
5	Total influência	Todas as práticas do processo sofrem influência.
4	Grande influencia	Mais que 50% das práticas do processo sofrem influência.
3	Média influência	50% das práticas processo sofrem influência
2	Baixa influência	Menos que 50% das práticas do processo sofrem influência
1	Não influência	Nenhuma das práticas do processo é afetada.

Fonte: Os autores

Após a atribuição de pontos baseada na escala Likert, foi possível apontar o grau de influência das práticas DevOps nos processos de gestão de TI selecionados do COBIT de acordo com o entendimento dos especialistas.

Conforme os processos de gestão escolhidos, o quadro 2 apresenta as práticas de gestão de cada um dos processos selecionados para a avaliação de influência do *DevOps*, conforme definido no manual específico do COBIT (ISACA, 2012a).

**Quadro 2 – Identificação das práticas de gestão dos processos COBIT selecionados para avaliação da influência do DevOps**

Práticas dos Processos COBIT		Processo COBIT selecionado	
		BAI03 - Gerenciamento da identificação e desenvolvimento de soluções	BAI04 - Gerenciamento da disponibilidade e capacidade
Estabelecer um plano de implementação.	BAI07.1	Projetar soluções em alto nível.	BAI03.01
Gerenciar mudanças de emergência.	BAI07.2	Projetar componentes da solução em detalhes.	BAI03.02
Planejar os testes de aceitação.	BAI07.3	Desenvolver os componentes da solução.	BAI03.03
Estabelecer o ambiente de teste.	BAI07.4	Adquirir componentes para a solução.	BAI03.04
Executar testes de desempenho.	BAI07.5	Construir as soluções.	BAI03.05
Promover para a produção e gerenciar versões.	BAI07.6	Desempenhar a garantia da qualidade.	BAI03.06
Fornecer suporte de produção inicial.	BAI07.7	Preparar para o teste da solução.	BAI03.07
Executar a revisão pós-implementação.	BAI07.8	Executar o teste da solução.	BAI03.08
Consolidar e facilitar uma cultura de compartilhamento do conhecimento.	BAI08.1	Gerenciar as mudanças nos requisitos.	BAI03.09
Identificar e classificar fontes de informação.	BAI08.2	Manter soluções.	BAI03.10
Organizar e contextualizar informação em conhecimento.	BAI08.3	Definir os serviços de TI e manter o portfólio de serviços.	BAI03.11
Organizar e contextualizar informação em conhecimento.	BAI08.4	Avaliar a disponibilidade, desempenho e capacidade atuais e criar um baseline.	BAI04.01
Avaliar e retirar informações.	BAI08.5	Avaliar o impacto no negócio.	BAI04.02
Identificar e registrar ativos atuais.	BAI09.1	Planejar para novos ou para mudanças nos requisitos do negócio.	BAI04.03
Gerenciar ativos críticos.	BAI09.2	Monitorar e revisar a disponibilidade e a capacidade.	BAI04.04
Gerenciar o ciclo de vida do ativo.	BAI09.3	Investigar e tratar questões de disponibilidade, desempenho e capacidade.	BAI04.05
Otimizar o custo dos ativos.	BAI09.4	Estabelecer um plano de implementação.	BAI07.1
Gerenciar licenças.	BAI09.5	Planejar a conversão dos processos de negócio, sistemas e aplicativos.	BAI07.2
Estabelecer e manter um modelo de configuração.	BAI10.1	Planejar os testes de aceitação.	BAI07.3
Estabelecer e manter um repositório e uma linha de base de configuração.	BAI10.2	Estabelecer o ambiente de teste.	BAI07.4
Manter e controlar os itens de configuração.	BAI10.3	Executar testes de desempenho.	BAI07.5
Produzir relatórios sobre o status e a configuração.	BAI10.4	Promover para a produção e gerenciar versões.	BAI07.6
Verificar e rever a integridade do repositório de configuração.	BAI10.5	Fornecer suporte de produção inicial.	BAI07.7
Executar procedimentos operacionais.	DSS01.1	Executar a revisão pós-implementação.	BAI07.8
Gerenciar serviços terceirizados de serviços de TI.	DSS01.2		
Monitorar a infraestrutura de TI.	DSS01.3		
Gerenciar o ambiente.	DSS01.4		
Gerenciar instalações.	DSS01.5		

Fonte: Adaptação de ISACA, 2012a

Na próxima seção é realizada a apresentação e a análise dos resultados encontrados.

#### 4. APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Com a avaliação da influência das práticas DevOps sobre os processos COBIT, através do teste de fáce, realizado com três especialistas em COBIT, chegou-se aos resultados apresentados e discutidos a seguir. Para melhor entendimento da análise realizada, é apresentada o resultado de cada prática DevOps.

Os gráficos apresentados na Figura 1 apresentam no eixo Y o nível de influência de acordo com a escala Likert apresentada na Tabela 2. Já o eixo X apresenta a escolha dos especialistas denominados como E1, E2 e E3, para cada processo COBIT analisado (BAI03, BAI04, BAI06, BAI07, BAI08, BAI09, BAI10 e DSS01), respectivamente.

Para a identificação do grau de influência percebido foi aplicada a escala da Tabela 2. Entretanto, para fins deste estudo, aplicou-se a regra de arredondamento para a mais ou para menos. Por exemplo, quando o resultado fôsse acima de 3,5 considerou-se como grande a influência, abaixo deste valor de média influência e assim sucessivamente com os demais

intervalos da escala. Dessa forma as Figura 1, 2 e 3 apresentam os resultados das avaliações em relação a todas as práticas DevOps analisadas, ou seja, ‘Entrega Contínua’, ‘Implantação Contínua’, ‘Integração Contínua’, ‘Infraestrutura como Código’ e ‘Monitoramento Contínuo’.

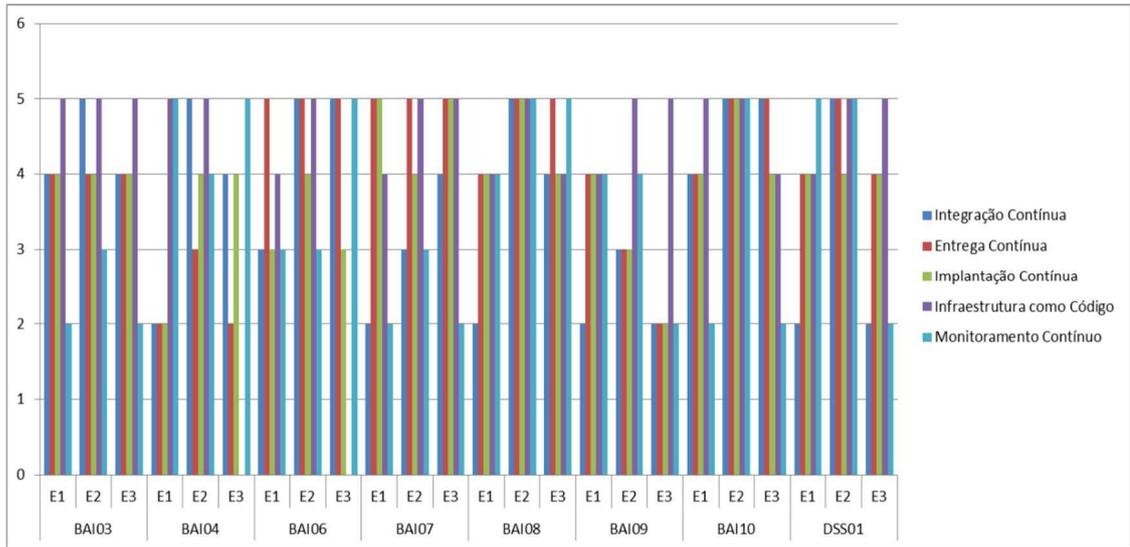


Figura 1: Resultado da avaliação da influência das práticas DevOps  
Fonte: Os autores

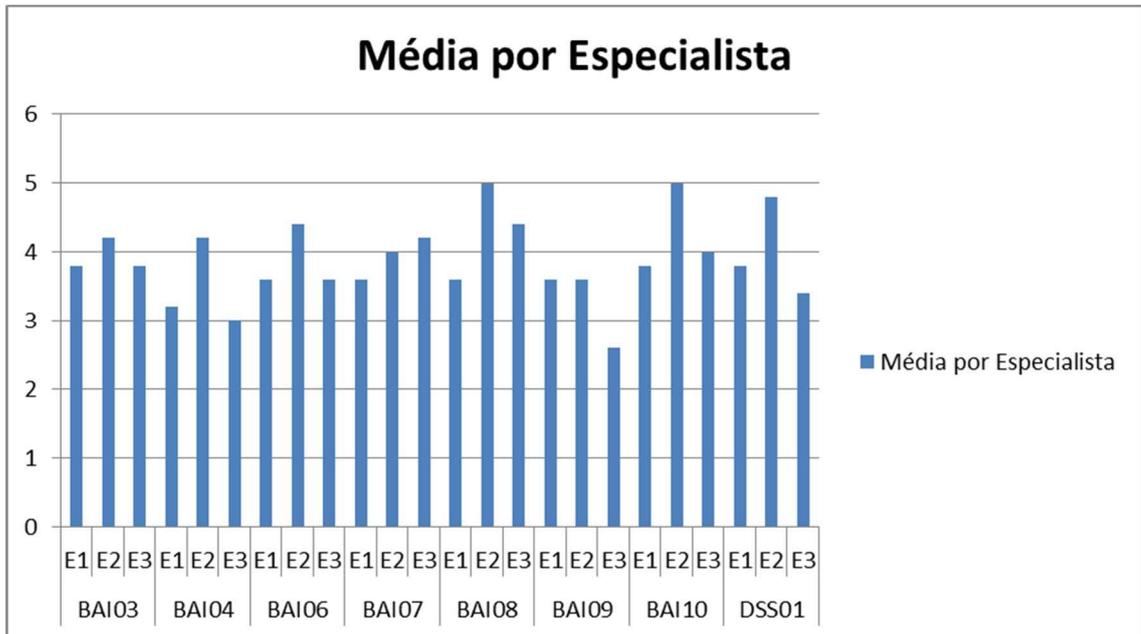


Figura 2 – Média dos resultados dos processos por especialista  
Fonte: Os autores

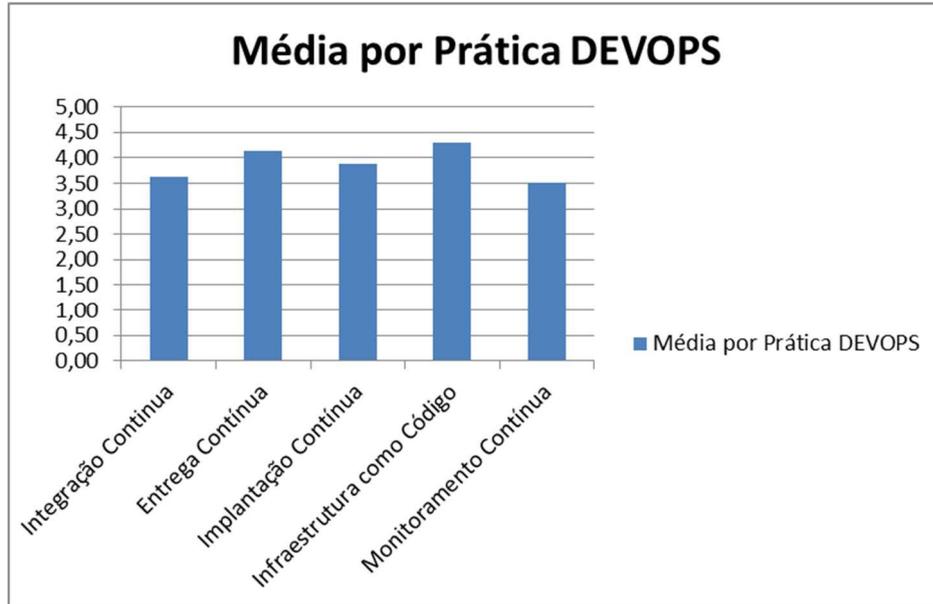


Figura 3 – Média geral dos resultados por prática DEVOPS

Fonte: Os autores

A prática DevOps de ‘Integração Contínua’, que foca no desenvolvimento de código e sua integração apresenta, de acordo com os resultados dos testes de face com os três especialistas em COBIT 5, uma grande influência sobre os processos BAI03, BAI04, BAI06, BAI08, BAI10, média influência nos processos BAI07 e DSS01 e baixa influência no processo BAI09. A ‘Integração Contínua’ atua fortemente no ciclo de desenvolvimento de uma solução, pois é onde o código é escrito e integrado ao software de forma contínua. O desenvolvimento de código ocorre no processo de gerenciamento da mudança, pois a cada novo código na mudança de um software tem que ter mecanismos de autorização, mesmo que com regras automatizadas no processo.

Um dos pilares do DevOps é justamente a gestão do conhecimento, onde as equipes de desenvolvimento e operação compartilham conhecimento entre si. Isto explica o fato do processo BAI08 ter demonstrado ser influenciado significativamente pela ‘Integração Contínua’. Em relação ao BAI10, gestão de configuração, também há grande influência, pois, o gerenciamento de versões e configuração do software é crítico para o processo. Já o processo BAI04 também sofre grande influência, pois é no desenvolvimento do software que os requisitos não funcionais do ambiente de operação são determinados e implementados pela equipe de operação concomitantemente com a equipe de desenvolvimento.

A influência média nos processos BAI07 e DSS01, respectivamente, gerenciamento da aceitação e da transição da mudança e gerenciamento de operações pode ser explicada pelo fato de que a ‘Integração Contínua’ foca no processo de construção e integração do código. Quanto a baixa influência sobre o processo BAI09 pode ser explicado porque este processo trata principalmente do gerenciamento de ativos como equipamentos e dispositivos de hardware. Visto de forma transversal, a influência da ‘Integração Contínua’ é significativa como mostra a Figura 3.

A práticas DevOps de ‘Entrega Contínua’ que foca na entrega do código pronto para ir para a revisão de qualidade ou testes de homologação de uma forma ágil, apresenta, de acordo com os resultados dos testes de face e com os três especialistas em COBIT 5, uma total influência nos processos BAI06, BAI07, BAI08 e BAI10 e grande influência nos processos BAI03 e DSS01, média influência nos processos BAI09 e baixa influência no processo BAI04. A ‘Entrega Contínua’ ainda faz parte do ciclo de desenvolvimento de um software, fato que explica sua total influência nos processos de gerenciamento de mudanças e no processo de

aceitação e transição da mudança, pois é quando o código integrado está pronto para ser entregue para a produção.

Em relação ao processo de gerenciamento do conhecimento como pilar do DevOps é fundamental o compartilhamento de informações sobre o código para todos os envolvidos no processo como um todo. No caso do processo de gerenciamento da configuração, pelo DevOps, o código somente é liberado para a operação quando está no repositório de gerenciamento da configuração. Já a grande influência dos processos de gerenciamento da identificação e desenvolvimento da solução e do gerenciamento de operações pode ser explicada pelo fato de que o processo de teste da solução ocorre no primeiro e procedimentos do ambiente de operações já devem estar definidos e implantados para que o software possa ser promovido para este ambiente através da prática DevOps de ‘Implantação Contínua’. A média influência do gerenciamento de ativos pode ser explicada em virtude deste processo ter um foco mais administrativo. Por fim o processo de gerenciamento da disponibilidade e capacidade sofre baixa influência da prática. Explica-se pelo fato do ambiente de operações ter sido dimensionado durante o desenvolvimento da solução, portanto antes desta prática ser acionada no *pipeline* de ‘Entrega Contínua’.

A prática DevOps de ‘Implantação Contínua’ que foca na publicação automatizada do software no ambiente de produção, também de uma forma ágil, parece ter grande influência sobre os processos BAI03, BAI07, BAI08, BAI10 e DSS01. Em relação ao BAI04, BAI06 e BAI09 parece apresentar média influência. A ‘Implantação Contínua’ parece influenciar o gerenciamento da identificação e desenvolvimento do software em virtude de que o processo de implantação contínua tem que ser desenvolvido e testado. No caso da publicação da aplicação para a produção a prática parece ter grande influência em todo o processo de gerenciamento da aceitação e transição da mudança que é o cerne da prática, a passagem do código para a operação. É importante observar que o gerenciamento do conhecimento também é influenciado pela necessidade de se compartilhar o status da versão que vai ser promovida para a operação no *pipeline* do processo. O processo BAI10 também é significativamente influenciado, pois quando o software é publicado deve atualizar o repositório da configuração, afeta o controle dos itens de configuração, no modelo e na avaliação do repositório. No processo DSS01 parece influenciar principalmente as práticas de procedimentos operacionais da produção, o monitoramento da infraestrutura que é uma faceta importante para o DevOps e o gerenciamento de serviços terceirizados, quando houver o uso de serviços na nuvem de terceiros. No tocante aos processos BAI04, BAI06 e BAI09 parece que tem média influência, considerando as respostas dos especialistas.

A prática de ‘Infraestrutura como Código’, de acordo com os especialistas, parece ter uma total influência nos processos BAI03, BAI07, BAI09, BAI10 e DSS01, grande influência no processo BAI08 e média influência nos processos BAI04 e BAI06. A ‘Infraestrutura como Código’ significa a entrega de uma infraestrutura ágil onde toda a configuração de máquinas e servidores já está pronta para uso de uma forma escalável e já preparada para a plataforma de software sobre a qual se está adicionando novas funcionalidades para o negócio. É compreensível que neste caso o gerenciamento da identificação e desenvolvimento de software é influenciado, pois a infraestrutura tem que ser desenvolvida como código. O gerenciamento da aceitação e da transição da mudança tem que estar praticamente no código que, automaticamente, vai fazer a transição para a infraestrutura. Entende-se isto como um utilitário que tem um processo que vai instalando o código automaticamente com pouca intervenção do profissional de operação. No caso do gerenciamento de ativos (BAI09) também parece ter grande influência. Isto talvez se explique pelo fato de que os ativos têm que estar administrados e disponibilizados para que a prática seja efetiva. Os ativos devem estar configurados adequadamente, não só *hardware*, como também o software, componentes e serviços previamente codificados e testados que fazem parte da infraestrutura. O processo de

gerenciamento das operações (DSS01) e também os procedimentos operacionais de produção e o monitoramento do ambiente estão dentro deste conceito da ‘Infraestrutura como Código’. O processo de gerenciamento do conhecimento (BAI08), diz que o processo tem que ser compartilhado entre as equipes de desenvolvimento e de operações. Esta prática, de acordo com os dados do estudo demonstra ter a maior influência transversal relativamente a todos os processos COBIT selecionados conforme mostra a Figura 3.

Por fim a prática de ‘Monitoramento Contínuo’, de acordo com os especialistas apresenta total influência sobre os processos BAI04, BAI08, grande influência sobre os processos BAI06 e DSS01, média influência sobre os processos BAI09, BAI10 e baixa influência sobre os processos BAI03 e BAI07. O ‘Monitoramento Contínuo’ realiza o monitoramento do ambiente de operação e do comportamento da aplicação a partir do momento que o código é promovido para a operação e, através do ‘Monitoramento Contínuo’, identifica necessidades de correções e de *back-out* para o ambiente de desenvolvimento, isto de forma muito rápida e contínua. Dessa forma o processo como o gerenciamento da disponibilidade e capacidade (BAI04) é tem que se ajustar aos requisitos do monitoramento contínuo. No caso do gerenciamento do conhecimento, os dados do monitoramento têm que estar disponíveis para todos os profissionais envolvidos no *pipeline*, pois qualquer *back-out* implica na reorganização do trabalho no processo como um todo. Em relação ao gerenciamento de mudanças (BAI06), o monitoramento contínuo afeta em virtude do processo de *back-out* o qual vai implicar no reinício do processo de gerenciamento da mudança do código que tem que ser refeito. No gerenciamento de operações (DSS01) é onde ocorrem os eventos que devem ser passados para o ‘Monitoramento Contínuo’ conforme requisitos previamente definidos. O gerenciamento de ativos e de configuração, BAI09 e BAI10, respectivamente, também são influenciados, pois os *logs* e eventos para o ‘Monitoramento Contínuo’ devem ter informações acerca do ativo e da configuração que está no ambiente de produção. Os demais processos parecem ter pouca influência sobre a prática.

## 5. CONSIDERAÇÕES FINAIS

Nesta pesquisa procurou-se estabelecer um entendimento preliminar sobre a influência das práticas *DevOps* sobre processos de gestão de TI do COBIT, especificamente os processos mais relacionados, no entendimento dos autores e da literatura, com as práticas *DevOps*. A análise feita a partir das práticas *DevOps* possibilitou a análise da influência sobre as práticas de gestão dos processos selecionados.

A escolha do modelo COBIT, permitiu uma análise aderente as melhores práticas de governança e gestão de TI utilizadas pelas corporações, bem como utilizar toda uma estrutura de processos que contém práticas de gestão de forma bem detalhada.

O uso da ferramenta MIDPTI permitiu estabelecer uma relação cruzada de processos e práticas *DevOps*, através de uma escala que apontava um grau de influência das práticas *DevOps* no estudo desenvolvido pelos autores.

Devido ao grande número de processos do modelo de referência do COBIT e de práticas respectivas de gestão, optou-se por analisar oito processos, mais especificamente: 1) BAI03 - Gerenciamento da identificação e desenvolvimento de soluções; 2) BAI04 - Gerenciamento da disponibilidade e capacidade; 3) BAI06 - gerenciamento da mudança; 4) BAI07 - gerenciamento da aceitação e transição da mudança; 5) BAI08 – gerenciamento do conhecimento; 6) BAI09 - gerenciamento de ativos; 7) BAI10 - gerenciamento da configuração e 8) DSS01 - gerenciamento das operações.

Verificou-se que os processos de gerenciamento do conhecimento (BAI08) e gerenciamento da configuração (BAI10) são os processos mais influenciados pelas cinco práticas *DevOps*, objeto do estudo, e que a prática de infraestrutura como código é a que mais influencia os processos selecionados. Os demais processos parecem ter pouca influência sobre a prática.

É importante ressaltar as limitações do estudo, visto que o modelo de referência COBIT 5 tem 37 processos e que no presente estudo foram selecionados apenas oito processos considerados mais aderentes às práticas *DevOps*, assim como foram utilizadas percepções de três especialistas as quais podem ter tido viés no entendimento dos conceitos e das práticas *DevOps*. O foco foi claramente a operação do *DevOps*, pois processos mais relacionados com a implantação de uma operação *DevOps* não fizeram parte do estudo.

Como contribuição deste trabalho aos pesquisadores sobre a temática apresentada, além de sua atualidade na academia, é a indicação, ainda que preliminar, da intensidade da influência das práticas *DevOps* sobre os processos do framework COBIT.

É necessário ir além, principalmente na especificação dos processos COBIT para a sua total aderência a um novo cenário de desenvolvimento de software num fluxo contínuo e de alta produtividade e qualidade e fortemente automatizado e com práticas onde há um alto nível de colaboração e de trabalho em equipe entre equipes de desenvolvimento e operações.

As pesquisas dos autores permanecem em andamento para buscar com maior profundidade os impactos das práticas *DevOps* na gestão de Tecnologia da Informação considerando uma abordagem de levantamento do tipo *survey*. Os resultados deste artigo podem também impulsionar outros trabalhos que relacionem o *DevOps* com níveis de capacidade de processos, integração com processos ITIL, a influência dos pilares do *DevOps* com a capacidade e desempenho da TI com *DevOps*, gerando métricas e metas que contribuam para o sucesso das organizações e a influência do *DevOps* com processos COBIT relativos ao domínio APO (Alinhar, planejar e organizar) pois é o que trata da implantação de estratégias e de arquiteturas.

## REFERÊNCIAS

- Audy, J. L. N., & Prikladnicki, R. (2007). *Desenvolvimento distribuído de software*. Rio de Janeiro: Elsevier.
- Braga, F. A. M. (2015). *Um panorama sobre o uso de práticas DevOps nas indústrias de software*. 2015.
- Duvall, P. M. (2011). *Continuous Delivery Patterns and AntiPatterns in the Software LifeCycle*. Dzone.
- Duvall, P. M., Matyas, S., & Glover, A. (2007). *Continuous Integration: Improving Software Quality and Reducing Risk*. Indianapolis: Addison-Wesley Professional.
- Erich, F., Amrit, C., & Maya, D. (2014). *DevOps Literature Review*. University of Twente, Report.
- Fernandes, A. A., & Abreu, V. F. (2014). *Implantando a governança de TI: da estratégia à gestão dos processos e serviços*. Rio de Janeiro: Brasport, 4. ed.
- Fowler, M. (2006). *Continuous Integration*. Disponível em: <<http://www.martinfowler.com/articles/continuousIntegration.html>>. Acesso em 08 set. 2016.
- Freire, F. (2013). *O que é DevOps?* EUA: IBM developerWorks. Disponível em: <[https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/o\\_que\\_devops?lang=en](https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/o_que_devops?lang=en)>. Acesso em 08 ago. 2016.
- Giampaoli, R. Z., Testa, M. G., & Luciano, D. M. (2011). Contribuições do modelo COBIT para a Governança Corporativa e de Tecnologia da Informação: desafios, problemas e benefícios na percepção de especialistas e CIOs. Porto Alegre: *Análise*, v. 22, n. 2, p. 120-133.
- Humble, J. (2010). *Continuous delivery vs continuous Deployment*. EUA: Continuous delivery. Disponível em <<http://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-Deployment>>. Acesso em 08 set. 2016.
- Humble, J., & Farley, D. (2013). *Entrega contínua: como entregar software de forma rápida e confiável*. São Paulo: Bookman.

- Huttermann, M. (2012). *DevOps for Developers: Integrate Development and Operations*. The Agile Way.
- ISACA. (2012a). *COBIT 5: Enabling Processes*. ISACA: Rolling Meadows.
- ISACA. (2012b). *Modelo corporativo para governança e gestão de TI da organização: COBIT 5 - Framework*. Rolling Meadows: ISACA.
- ISACA. (2015). *DevOps practitioner considerations*. ISACA: Rolling Meadows.
- Kim, G. (2013). *Top 11 things you need to know about DevOps*. IT Revolution Press. Disponível em: <http://miroslawdabrowski.com/downloads/DevOps/Top%2011%20Things%20You%20Need%20To%20Know%20About%20DevOps.pdf>. Acesso em 11 set. 2016.
- Lunardi, G. L., Dolci, P. C., Maçada, A. C. G., & Becker, J. L. (2014). Análise dos mecanismos de governança de TI mais difundidos entre as empresas brasileiras. *Alcance*, 21(1).
- Magalhães, F. L. F., Gaspar, M. A., Costa, I. & Campos, J. G. F. (2017). Planejamento estratégico de tecnologia da informação: análise de conceitos, frameworks e processos apresentados em livros publicados no Brasil. *Espacios (Caracas)*, 38(1), p. 31.
- Palko, T. (2015). *Monitoring in the DevOps Pipeline*. SEI - Technical Guidelines and Practical Advice for DevOps. Disponível em: <<https://insights.sei.cmu.edu/devops/2015/12/monitoring-in-the-devops-pipeline.html>>. Acesso em 27 abr.2017.
- Rappaport, R. (2014). *A short history of Devops*. CA Technologies. Disponível em: <<http://www.ca.com/us/rewrite/articles/devops/a-short-history-of-devops.html>>. Acesso em 11 set. 2016.
- Reddy, A. (2013). *DevOps: The IBM Approach*. Nova York: IBM Corporation.
- Riley, C. (2015). *Métricas de DevOps*. *DevOps*. Disponível em: < <https://devops.com/metrics-devops/>>. Acesso em 20 abr. 2017.
- Ross, J. W. *et al.* (2016). *Design digital organizations*. Center for information systems research. MITSloan. Working Paper, n. 406.
- Sato, D. (2013). *DevOps na Prática: Entrega de Software Confiável e Automatizada*. São Paulo: Casa do Código.
- Selig, G. J. (2016). IT Governance - an integrated framework and roadmap: how to plan, deploy and sustain for improved effectiveness. *Journal of International Technology and Information Management*. 25(1), p. 55-76.
- Sharma, S. (2014). *DevOps for Dummies*. New Jersey: IBM Limited Edition, John Wiley & Sons, Inc., Hoboken.
- Taurion, C. (2014). *Você realmente conhece o conceito de DevOps?* TI Especialistas, 23. abr. 2014. Disponível em: <<http://www.tiespecialistas.com.br/2014/04/voce-realmente-conhece-o-conceito-de-devops>>. Acesso em 05 jan. 2016.
- Taurion, C. (2016). *O primeiro passo: transformação digital como base para os negócios do século 21*. São Paulo: Amazon.
- Wahaballa, A., Wahaballa, O., Abdellatief, M., Xiog, H., & Quin, Z. (2015). *Toward Unified DevOps Model*. In: VI IEEE International Conference on Software Engineering & Service Science, IEEE, p. 211-214.
- Wettinger, J., Andrikopoulos, V., & Leymann, F. (2015). *Automated Capturing and Systematic Usage of DevOps Knowledge for Cloud Applications*. IEEE Computer Society: IEEE International Conference in Cloud Engineering. IEEE Computer Society, p. 60-65.
- Zentgraf, D. (2013). *Definindo a Implementação Distribuível em DevOps*. IBM Developer Works. Disponível em: <<http://www.ibm.com/developerworks/br/rational/library/defining-deployment-deliverable-devops/index.html>>. Acesso em 13 dez. 2016.