Automated Technology Transfer in MDE as a Service: Experiences and Research Directions

Fábio Paulo Basso[1], **Diego Luiz Kreutz**[1], **Elder de Macedo Rodrigues**[1], **Maicon Bernardino**[1]

[1]Laboratory of Empirical Studies in Software Engineering (LESSE), Federal University of Pampa (UNIPAMPA), Alegrete, RS, Brazil

{fabiobasso,diego.kreutz,elderrodrigues}@unipampa.edu.br,bernardino@acm.org

Abstract. *For years, researchers have sought to assist in the execution of Software Engineering (ES) activities with toolboxes developed in the laboratory and in-house. Big market leaders like IBM, Google and Oracle embrace these software development assistants innovations. Unfortunately, innovations launched by independent researchers simply cannot get space and are shelved. Due to the lack of maturity of the product, there is a difficulty, in less than three years, that it is the threshold for success or failure of a startup in Brazil, to enable the transfer of these technologies from academia to industry. This paper presents some experiences with a future vision for research that seeks to automate the transfer of technology in the area through integration mechanisms for assets. In particular, we discuss a set of works produced in this direction, highlighting the potential of distributed architectures, which can be built with technologies for smart contracts and blockchain, in gaining credibility within short time-to-market.*

Resumo. *Há tempo os pesquisadores vêm tentando apoiar a execução de atividades da Engenharia de Software (ES) com suporte ferramental desenvolvido em laboratório e in-house. Líderes de mercado como IBM, Google e Oracle abraçam estas inovações de base tecnológica para ES e, infelizmente, ainda se observa que inovações lançadas por pesquisadores independentes enfrentam muitas dificuldades para conseguir espaço. Devido à falta de credibilidade do produto por sua imaturidade, é difícil nos primeiros três anos determinantes ao sucesso ou fracasso de uma startup no Brasil, se viabilizar a transferência de tecnologias da academia para a indústria. Neste cenário, o presente artigo apresenta algumas experiências com uma visão de futuro para pesquisas que buscam automatizar a transferência de tecnologia da área por meio de mecanismos de integração para assets. Em especial, discute-se um conjunto de trabalhos produzidos neste rumo, destacando o potencial de arquiteturas distribuídas, as quais podem ser construídas com tecnologias como smart contracts e blockchain, no ganho de credibilidade dentro do curto tempo de mercado.*

1. Introdução

Atualmente, a sociedade brasileira impõe pressão para que as pesquisas conduzidas na academia tenham aplicação na prática. Hoje se questiona se de fato, como pesquisadores, estamos prontos para responder positivamente a esta pressão. Nesse cenário, se

percebe que boa parte das pesquisas conduzidas na área de Engenharia de Software tem um cunho prático e são realizadas por meio de estudos aplicados [Sommerville 2010]. No entanto, são raros os relatos de transferência de tecnologia observados na literatura desta área [Mikkonen et al. 2018], o que é algo bastante controverso. Para uma área de pesquisa da computação construída nos estudos aplicados, é necessário entender os motivos que atrapalham a transferência de tecnologia.

Há anos, pesquisadores buscam apoiar a execução de atividades de Engenharia de Software com suporte ferramental desenvolvido em laboratório e *in-house* [Mussbacher et al. 2014]. Empresas como IBM, Google e Oracle abraçam as inovações de base tecnológica para ES, o que é positivo para algumas startups de base tecnológica que tiveram a competência e sorte de convencê-los como investidores. Porém, isto também deixa pouco espaço para pesquisadores independentes. Por exemplo, Flutter é um um toolkit bastante adotado em fábricas de software para desenvolvimento de apps[1]. Este toolkit é mantido pela Google, oferece uma Domain Specific Language (DSL) [France and Rumpe 2007] para desenvolvimento e permite a geração de código para múltiplas plataformas. O fato interessante associado com a transferência de tecnologia de ES é que, antes de 2000 [France and Rumpe 2007], muitas DSLs e toolkits foram concebidas com o mesmo propósito do Flutter, mas como parte de pesquisas em Model-Driven Engineering (MDE) [Mussbacher et al. 2014]. No entanto, sua expressiva adoção por fábricas de software não foi observada até então [Basso 2017].

Segundo estudo recente executado pelo Sebrae, uma em cada quatro start-ups brasileiras fecha antes de completar dois anos[2]. A julgar pelo período médio dedicado por incubadoras de base tecnológica como a RAIAR e Incubadora PampaTec Alegrete para as startups, as novas ideias têm menos de três anos para provar o seu valor ao mercado. Isso requer agilidade para lançar o produto no mercado e para se obter o máximo de reputação positiva para este produto em um curto espaço de tempo. Logo, é preciso romper com o *modus operandi* da transferência de tecnologia, uma vez que é impossível adquirir reputação para tecnologias desenvolvidas *in-house* para assistir processos de desenvolvimento de software, como ferramentas de produção construídas para MDE, em tão pouco tempo. Portanto, para pesquisadores independentes e *start-ups*, aqui entra o grande desafio: atender ao tempo de mercado, uma vez que além de desenvolver o produto com qualidade é também preciso obter reputação para que o mesmo se torne visível para o público alvo.

Em um trabalho anterior, constatou-se que uma dificuldade está em fazer os ativos de tecnologia alcançarem o mercado [Neto et al. 2019], propondo ecossistemas como um meio de se conseguir algumas facilidades, como busca de concorrentes e estabelecimento de parcerias na comercialização e integração de ativos. Para a área de Engenharia de Software, a nossa definição de ativos, ou *asset*, inclui ferramentas de modelagem ou de refinamento de modelos, bem como qualquer recurso que dê suporte para atividades da Engenharia de Software por meio de modelos [Basso 2017]. Tais *assets* visam aumentar a qualidade dos produtos de software, assim como automatizar atividades como a assistência para representação arquitetural do modelo do software [Basso et al. 2016] e desenvolvimento de software baseado em modelos e geração de código [Basso et al.

[1] www.flutter.dev

[2] https://exame.abril.com.br/negocios/dino/1-a-cada-4-empresas-fecha-antes-de-completar-2-anos-no-mercado-segundo-sebrae/

2013]. Portanto, independentemente da natureza do *asset* e de seus criadores (indústria ou academia), tem-se como meta introduzi-lo em fábricas de software, sendo o suporte ferramental algo essencial para o sucesso neste tipo de serviço.

Esse artigo aborda a transferência de tecnologia por meio de serviços especialistas, denominados por MDE como um Serviço (MDEaaS - *MDE as a Service*) [Basso et al. 2017a]. MDEaaS inclui a execução de fases de integração de ativos em cadeias de ferramentas (*toolchains*) e processos que atendem contextos específicos de fábricas de software operando em cenários de coopetição, isto é, cenários que envolvem a colaboração entre empresas competidoras [Ritala et al. 2014]. Nestes serviços, cadeias de ferramentas são customizadas e processos automatizados para cada caso de uma empresa prospectada para transferência de tecnologia. Outra característica peculiar no MDEaaS é que cadeias de ferramentas e processos incluem ativos híbridos, geralmente ferramentas produzidas por mais de um indivíduo/empresa em contextos inter-organizacionais [Basso 2017]. Como operaram sobre negociações inter-organizacionais, estes serviços precisam considerar uma perspectiva tridimensional de dados [Neto et al. 2019], os quais incluem visões técnicas, sociais e de negócios como fontes de influência na tomada de decisão.

Nesse contexto, o presente artigo destaca os benefícios que podem ser alcançados com o uso de tecnologias como *smart contracts* e *blockchains*, que fornecem mecanismos para se obter confiança nos ativos produzidos por desenvolvedores independentes, em curto tempo de mercado. O artigo está organizado da seguinte forma: a Seção 2 relata as experiências no MDEaaS; a Seção 3 apresenta lacunas de pesquisa; e, por fim, a Seção 4 apresenta as conclusões.

2. Experiências no MDE como um Serviço

A transferência de tecnologia na Engenharia de Software caracteriza-se pela transferência de conhecimento técnico ou científico para produção de software. Um dos elementos centrais da transferência de tecnologias é elencar os fatores de produção associados com uma determinada tecnologia, resultantes de pesquisas e investigações científicas. Fatores de produção, como a velocidade de entrega de histórias de usuário por iteração do processo de desenvolvimento e número de defeitos detectados e corrigidos, são sempre considerados pelas fábricas de software em processos de tomada de decisão por determinadas tecnologias. Estes fatores são muito associados com a adoção de tecnologias introduzidas no mercado como forma de melhoria da qualidade e produtividade de software. Sendo assim, o desconhecimento desses fatores ocasiona um dos principais problemas relatados pela indústria na adoção de tecnologias de MDE: a falta de dados para investir em tecnologias produzidas *in-house*; dados estes que convençam candidatos à adoção de tecnologias para automação de tarefas do processo de desenvolvimento.

A transferência dessas tecnologias da área ainda é realizada de modo *ad-hoc*. Em pesquisas associadas com o MDEaaS, buscou-se tornar isto algo mais sistemático e assistido por ferramentas. Dessa forma, o MDEaaS pode ser uma possível solução para a automação na transferência de tecnologias em negócios de ES. Ele é visto como um nicho de negócio por algumas empresas, indivíduos atuando como profissionais liberais, bem como visionários que buscam criar suas primeiras empresas de serviços na área, e é caracterizado por uma demanda de empresas que contratam tais profissionais para a introdução de tecnologias para MDE em contextos de fábricas de software. Sob uma perspectiva pu-

ramente técnica, tal demanda inclui a automação de uma ou mais atividades da Engenharia de Software, usando-se um conjunto de ativos de MDE. Por exemplo, DSLs para representação de requisitos de software em modelos, representação de processos de negócio e de software, bem como ativos que transformam tais modelos em formatos interpretados por ferramentas utilizadas ao longo de toda a cadeia produtiva do software.

Por meio das experiências anteriormente citadas, prospectou-se que MDEaaS é de grande relevância para a automação da transferência de tecnologia em geral e pode promover benefícios ainda não tangíveis para as inovações da área como: 1) a possibilidade de realizar colaborações/negociações casadas de ativos, o que requer autenticação das partes interessadas na composição de cadeias de ferramentas produzidas de modo independente de provedor/líder de tecnologia; 2) a abertura de novos negócios por meio de ativos representados e armazenados na *deep-web*, como *assets* distribuídos e sem a presença de um líder no processo; 3) ruptura total na forma como se aborda credibilidade em relação a ativos, por meio de uma relação mais direta entre fábricas de software e fornecedores de tecnologias.

O foco do MDEaaS não está na modelagem dos sistemas e processos, mas no serviço fornecido por um Engenheiro de Software na transferência de tecnologia de MDE para um contexto de adoção. Logo, um portfólio de serviços pode incluir a automação de diversas atividades em muitas fases de processos de desenvolvimento de software. Essa automação atualmente é custosa e sua implementação depende de uma equipe interna de uma fábrica de software que domina profundamente um conjunto bem limitado de tecnologias. Como geralmente não há uma especialidade dentro da fábrica de software para tratar disso, a automação é desencorajada. Já os profissionais no MDEaaS devem ser especialistas na transferência de tecnologias de MDE. Eles devem dominar múltiplas tecnologias de MDE (ativos), que têm muitas aplicações nas fases de processos de desenvolvimento de software. Portanto, MDEaaS pode ser a solução que a indústria de software busca na automação de seus processos.

MDE como um Serviço trata da introdução de ativos de MDE em contextos inter-organizacionais, isto é, quando mais de uma empresa está prospectada para adoção desses ativos. Para exemplificar estes contextos, parte-se de um pequeno relato de implantação de um negócio focado no MDEaaS: uma *spin-off* de mestrado provedora de serviços de transferência dessas tecnologias. Para dar suporte para estes serviços, em 2006 propôs-se a FOMDA DSL, uma ferramenta para tratar de questões técnicas na transferência de tecnologia de MDE. Em seguida, criou-se uma *start-up*, chamada Adapit, que operou de 2007 até 2011 em um projeto para a implementação do MDEaaS. Por meio dessa *start-up*, foi possível gerenciar semelhanças e diferenças de três contextos inter-organizacionais. Em termos técnicos, fez-se a transferência de tecnologia de MDE pela representação de reutilização em ativos de MDE, em uma abordagem de reúso sistemático assistida por ferramenta que implementa conceitos de Linha de Produtos de Software, Desenvolvimento Baseado em Componentes, Cadeia de Transformação de Modelos e Testes Adaptativos. A partir dessa representação, o Engenheiro de Software adapta automaticamente os ativos para os contextos de adoção de tecnologias.

Essa experiência mostrou que, apesar dos conhecimentos da Engenharia de Software serem aplicáveis ao MDEaaS, diversos desafios técnicos e metodológicos precisavam ser resolvidos para viabilizar tais serviços. Assim, de 2012 até 2017, investigou-se

um desses problemas: uma abordagem para reúso oportunista, construída sobre uma representação comum para *assets*, que almeja viabilizar a colaboração e competição (coopetição) entre empresas de desenvolvimento de software como um desafio na transferência de tecnologias de MDE. Coopetição é caracterizada pela colaboração entre empresas competidoras como, por exemplo, duas empresas A e B, que desenvolvem software embarcado, cada qual atendendo e mantendo a sua base de clientes isoladamente. Elas são competidoras em seus portfólios. Mas, por não possuírem especialistas e, para dividir os custos na implantação de ativos de MDE, pretendem colaborar na automação de seus processos. Nesse sentido, o MDEaaS é uma possível solução, em que o profissional atende ambas as empresas e amortiza os custos na transferência de tecnologias.

Em nossos estudos preliminares, se ofereceu soluções voltadas para problemas técnicos na implementação do MDEaaS [Basso et al. 2013, Basso et al. 2015, Basso et al. 2016, Basso et al. 2017b]. Porém, estas contribuições não tratam adequadamente de questões mais voltadas para negócio e também não ajudam com critérios de informações que se deve prover aos possíveis clientes para a tomada de decisão na transferência de tecnologia como composição de sistemas. Assim, uma vez que se progrediu nas questões técnicas que fundamentam uma representação comum para tais tecnologias, deve-se também tratar de desafios técnicos e metodológicos envolvendo a aquisição de tecnologias como ativos negociados em cadeias de ferramentas.

Transferir uma tecnologia de Engenharia de Software para um determinado contexto parece uma atividade trivial, no entanto, isto provou-se bastante desafiador na teoria e na prática. Na prática, a dificuldade para transferência de tecnologia está em tornar a inovação atrativa para um grande provedor de tecnologia, a ponto deste adotá-la. Esta dificuldade se deve a um modelo de negócio ruim, baseado em competição, em vez de uma colaboração com os grandes provedores de tecnologia da área, como Google, IBM e Oracle. É muito difícil convencer estas empresas a tornarem-se colaboradoras, então elas são tratadas como concorrentes para aquilo que se produz em laboratório ou em *in-house*. Isso é ruim para a inovação tecnológica, pois mata os produtos e serviços já na raiz.

Falta uma visão mais mercadológica daquilo que se produz em ambiente acadêmico. Por exemplo, serviços de transferência de tecnologia envolvendo integração de ferramentas em cadeias de ferramentas são atualmente realizados de modo *ad-hoc*, ou seja, sem uma boa fundamentação para sua execução, tanto processual quanto implementacional, que inclua a visão de negócios. Isso gera desconfiança do mercado potencial. Neste sentido, as duas principais limitações, não relacionadas com suporte ferramental, que hoje atrapalham a automação da transferência de tecnologia na área são:

1 - Faltam metodologias que sistematizam a transferência de tecnologia da área. Devido à existência de poucos relatos de experiência, percebe-se uma limitação na literatura para sistematizar a transferência de tecnologia de Engenharia de Software para MDE. Logo, não se tem conhecimento de uma abordagem sistemática que possa ser repetida diversas vezes para diferentes contextos de empresas, muito menos uma que considere negociações por meio de contratos inteligentes que garanta a segurança nas transações e a validade na integração dos ativos. Esta falta de uma sistematização se deve muito às limitações do corpo de conhecimento empírico na área, o que gera incerteza de sucesso na implementação do MDEaaS e aumenta os riscos na transferência dessas tecnologias para as fábricas de software.

2 - Fatores de produção associados com ativos da área são desconhecidos. Em análises do estado da arte anteriores, percebe-se a falta de um corpo de conhecimento para fomentar tais iniciativas na área. Por exemplo, não há contribuições científicas que auxiliam no mapeamento dos riscos do MDEaaS por meio de *blockchains*, nem das boas práticas para a condução destes serviços. Também inexiste uma abordagem sistemática de suporte ao Engenheiro de Software para definir fatores de produção, como a análise e prospecção da curva de aprendizado derivada de transferências casadas de tecnologia, melhoria/prejuízo de produção e o Retorno de Investimento (ROI) associado com os ativos de MDE. Fatores de produção são sempre avaliados pela indústria de software antes de investir na adoção de qualquer tecnologia, mas pouco se entende desses fatores no que tange serviços de Engenharia de Software em geral.

Já em relação ao suporte ferramental, nossa contribuição anterior trata de algumas lacunas de pesquisa proeminentes [Neto et al. 2019]. A próxima seção é complementar e trata da transferência casada de tecnologia usando-se arquiteturas distribuídas de ativos.

3. Lacunas de Pesquisa

Em experiências anteriores, buscou-se, por exemplo, o desenvolvimento de outras inovações em laboratório, incluindo DSLs e geradores de código para *Internet of Things* (IoT) por meio de redes de sensores sem fio [Paulon et al. 2014]. Na prática, percebe-se que a dificuldade não está em conceber o produto, mas em conseguir espaço no mercado, fornecendo credibilidade. Aprendeu-se que, sendo uma empresa pequena, assim como são hoje também os grupos de pesquisa das universidades, é impossível comercializar algo para competir com grandes provedores de tecnologia para desenvolvimento de software. Eles ditam as regras, ditam o que é bom e o que é ruim e, como consequência, por mais potencial que tenham alguns produtos e serviços, eles acabam engavetados. Neste sentido, busca-se promover uma disruptura no *modus operandi* da transferência de tecnologias da área de Engenharia de Software, que pode ser realizada por meio da integração do MDEaaS com contratos inteligentes e *blockchain*.

Contratos inteligentes (*smart contracts*) são protocolos de transação computadorizada que executam em termos de um contrato [Szabo 1996]. Os contratos inteligentes vêm sendo largamente utilizados e difundidos no contexto de *blockchains* e criptomoedas [Christidis and Devetsikiotis 2016]. No contexto de *blockchains*, os contratos inteligentes são *scripts* armazenados no *blockchain*, que possuem um endereço único e são acionados por meio da associação de uma transação. O contrato é, então, independente e segue, automaticamente, instruções pré-definidas em todos os nós da rede.

Tipicamente, em *blockchains* de criptomoedas, os contratos inteligentes são utilizados para definir funções de depósito e negociação de criptomoedas. De forma resumida e simplificada, um usuário A cria o contrato e deposita unidades de X no contrato. A função de negociação estabelece que 1 unidade de X equivale a 5 unidades de Y. Assim que o usuário A receber 5 unidades de Y de um usuário B, o contrato liberará 1 unidade de X para B e 5 unidades de X para A. Um contrato inteligente possui também uma função de "cancelamento", que permite ao usuário A "reverter" uma negociação e remover as suas unidades X do contrato.

Todas as negociações utilizando este tipo de contrato ocorrem entre partes mutuamente não confiáveis, o que é uma das características mais interessantes e potencial-

mente disruptivas de contratos inteligentes e *blockchains*. Não é para menos que hoje os contratos inteligentes são divulgados em múltiplos domínios do conhecimento como ferramentas confiáveis para eliminar intermediários e estabelecer negociações de ativos ou qualquer coisa de valor. Como exemplo, podemos utilizar um contrato inteligente para garantir um acordo de nível de serviço, em que caso o acordo seja cumprido, uma recompensa é dada, ou então uma penalidade, em caso contrário (podendo ou não usar criptomoedas).

Um dos desafios para integrar estas três novidades é identificar os requisitos de armazenamento de informação, nos blocos da *blockchain*, e a definição de funções para contratos inteligentes no escopo de negociação de ativos, no que se refere ao casamento de informações sobre ferramentas de produção de software em nível semântico e sintático, também com autenticação das partes envolvidas no estabelecimento de cadeias de ferramentas (cadeias de ferramentas). Assim, a tecnologia proposta deve permitir ao Engenheiro de Software gerenciar todo o processo de integração, hoje executado manualmente na automação de processos de desenvolvimento de software. Isso requer a execução de pesquisas focadas e de longa duração, realizadas por pesquisadores com diferentes visões e experiências na transferência de tecnologia.

Por mais revolucionária que seja a palavra "blockchain", sua aplicação em outros cenários, que não criptomoedas, ainda não está consolidada e diversas pesquisas são necessárias até ela se tornar estabelecida. Existe a perspectiva que isto só venha a acontecer entre 5 e 10 anos, quando *blockchain* deve atingir o platô da produtividade, segundo o Gartner [K. Panetta 2018]. Neste sentido, a aplicação de tecnologias para contratos inteligentes como um meio de promover automaticamente a transferência de tecnologia de MDE em contextos de Engenharia de Software não é tratada pela literatura, mas altamente desejável para a cadeia produtiva, visão também compartilhada pelo Gartner [Gartner 2013].

Embora as implementações mais conhecidas de *blockchain* são em criptomoedas, onde os blocos da cadeia armazenam conjuntos de transações que guardam os registros de movimentações de valores específicos, abre-se espaço para investigar um problema aplicado aos cenários de negócios da engenharia de software. Além disso, os estudos preliminares no MDEaaS focaram no desenvolvimento de tecnologias e no levantamento de fatores de produção que afetam a transferência destas inovações para contextos de fábricas de software por meio de cadeias de transformação de modelos. Prospecta-se que para uma aplicação em transferência automática de tecnologias, novas pesquisas utilizarão da temática para desenvolver novas plataformas distribuídas de *assets* com base em *blockchains* e contratos inteligentes. A expectativa é que o produto final desta integração viabilize e potencialize negociações derivadas da pesquisa científica, inicialmente, em Engenharia de Software, viabilizando a rápida disseminação de *assets* desenvolvidos em laboratório e *in-house*, por meio de contratos inteligentes e negociações par-a-par. Portanto, promoverá um meio alternativo para se obter credibilidade na transferência de tecnologias, uma vez que negociações se dariam entre entidades não confiáveis, isto é, eliminando a necessidade de entidades intermediadoras (os grandes líderes).

Blockchain, com sua segurança e confiabilidade, garante que a transação na transferência de tecnologia tenha ocorrido e não possa ser alterada, ou seja, trata-se de uma tecnologia que permite, hoje, a condução de estudos aplicados. O código da *blockchain*

automatiza as próximas etapas de transações feitas pelos participantes de um bloco de negócio, o que, em nível técnico, significa estabelecer cadeias de ferramentas como colaborações/vendas casadas entre múltiplas empresas. Assim, novas pesquisas podem investigar a aplicação do que já se tem fundamentado em termos de MDEaaS, para estabelecer negócios implementados por meio de contratos inteligentes.

Atualmente, por meio de integração (semi-)automática de ativos [Basso 2017], é possível divulgar inovações em repositórios de *assets*, estabelecendo, por meio de plataformas de *assets*, oportunidades para vendas, como produtos e serviços. Contratos inteligentes podem ser explorados neste sentido. Além disso, tais tecnologias estão sendo integradas de modo facilitado, quando construídas sobre o guarda-chuva de MDE. Ou seja, os ativos que podem ser negociados via contratos inteligentes são ativos elaborados com tecnologias de suporte ao MDE, mas de qualquer natureza para processos de desenvolvimento de software, e que precisam ser introduzidos em contextos de fábricas de software em negociações que envolvem a transferência de tecnologia entre múltiplas empresas. Logo, a tecnologia de *blockchain* pode promover uma ruptura em como essas negociações e integrações podem ser realizadas, necessitando investigações em termos de metodologias, ferramentas/plataformas e de formação de base de conhecimento na área.

Por fim, uma vez que se tem por objetivo de longo prazo estabelecer as fundamentações para a ruptura nos negócios envolvendo a transferência de tecnologia, conclui-se que tecnologias para *smart contract*, *blockchain* e MDEaaS devem ser investigadas de modo integrado. Em se tratando da transferência de tecnologia pra a área foco, cujo artigo foi divulgado, assume-se que o foco de estudos futuros são tecnologias específicas de Engenharia de Software, livres ou comerciais. As pesquisas de Engenharia de Software têm, até então, desenvolvido tais tecnologias de forma isolada, sem preocupação com sua transferência para a indústria de software. É hora de promover mudanças.

4. Conclusões

Como produtores de tecnologias que auxiliam nas atividades de Engenharia de Software (ES), a dificuldade dos pesquisadores dessa área está em fazer seus inventos adquirirem um status de ativos de valor comercial. Por meio de experiências em uma *start-up* concebida para esta finalidade, percebeu-se que outras limitações também atrapalham a transferência destas tecnologias (generalizadas como ativos): o corpo de conhecimento escasso que fornece suporte metodológico ao Engenheiro de Software e a falta de suporte ferramental que viabilize uma integração semi-automática desses ativos em processos de fábricas de software do mundo real. A dificuldade para se progredir nessa temática está, principalmente, na novidade do tema. O corpo de conhecimento em Engenharia de Software carece de estudos empíricos que forneçam as bases para computação dos fatores de produção associados com tecnologias de MDE e não fornece uma base para se implementar vendas casadas de serviços em negociações que ocorrem em cenários inter-organizacionais.

É preciso fundamentações para a transferência de tecnologia da área. Percebe-se também que não há uma abordagem sistemática para o MDEaaS que possa ser repetida diversas vezes para diferentes contextos de empresas. Esta falta de uma sistematização se deve muito às limitações do corpo de conhecimento empírico na área, o que gera incerteza de sucesso na implementação do MDEaaS e aumenta os riscos na transferência do que se

produz na academia, como tecnologias, para as fábricas de software. Portanto, sem essas bases, a transferência de tecnologia continuará uma tarefa *ad-hoc*, propensa a erros e com um grande risco associado na oferta e contratação destes serviços pelos clientes.

Neste sentido, pesquisas em *blockchain* e contratos inteligentes (*smart contracts*), em conjunto com uma linguagem de representação comum para ativos denominada RAS++ [Basso 2017] (construída sobre o guarda-chuva de MDE), oferecem uma solução promissora. Tais tecnologias podem, por exemplo, promover benefícios ainda não tangíveis para o cenário, como: 1) a possibilidade de realizar colaborações/negociações casadas de ativos, o que requer autenticação das partes interessadas na composição de cadeias de ferramentas produzidas de modo independente de provedor/líder de tecnologia; 2) a abertura de novos negócios por meio de ativos representados e armazenados na *deep-web*, como *assets* em repositórios distribuídos e sem a presença de um líder no processo; e 3) a ruptura total na forma como se aborda credibilidade em relação a ativos, por meio de uma relação mais direta entre fábricas de software e micro-fornecedores de tecnologias.

Por fim, vale ressaltar que uma solução para a transferência de tecnologia não é meramente uma questão de oferecer um suporte ferramental, como as de foco de investigação neste artigo, mas que envolve uma mudança cultural do pesquisador. Observa-se que na pesquisa brasileira, pouco se faz para ter um produto de mercado, pois se foca em protótipos. Além disso, existe um viés na academia em que pesquisadores criam tecnologias muito incipientes, muitas vezes com o objetivo de publicar artigos e não exatamente de fomentar a literatura com dados que auxiliem na transferência de tecnologia. Relacionado com a divulgação da tecnologia, repositórios abertos como ReMoDD e SEMAT, que recomendam o compartilhamento de diversos assets úteis para a indústria de software, infelizmente não são muito explorados. Já alternativas interessantes para publicação de material em revistas que podem ser utilizadas com este propósito, como Data in Brief, MethodsX e SoftwareX, começam a ganhar notoriedade.

Referências

[Basso 2017] Basso, F. P. (2017). *RAS++: Representing Hybrid Reuse Assets for MDE as a Service. Av at <www.cos.ufrj.br/uploadfile/publicacao/2811.pdf>*. PhD thesis, Universidade Federal do Rio de Janeiro (UFRJ).

[Basso et al. 2017a] Basso, F. P., Oliveira, T. C., Werner, C. M., and Becker, L. B. (2017a). Building the foundations for 'mde as service'. *IET Software*, 11:195–206(11).

[Basso et al. 2013] Basso, F. P., Pillat, R. M., Oliveira, T. C., and Becker, L. B. (2013). Supporting large scale model transformation reuse. In *12th International Conference on Generative Programming: Concepts & Experiences.*, GPCE'13, pages 169–178.

[Basso et al. 2016] Basso, F. P., Pillat, R. M., Oliveira, T. C., Roos-Frantz, F., and Frantz, R. Z. (2016). Automated design of multi-layered web information systems. *Journal of Systems and Software*, 117:612 – 637.

[Basso et al. 2015] Basso, F. P., Pillat, R. M., Roos-Frantz, F., and Frantz, R. Z. (2015). Combining mde and scrum on the rapid prototyping of web information systems. *International Journal of Web Engineering and Technology*, 10(3):214–244.

[Basso et al. 2017b] Basso, F. P., Werner, C. M. L., and de Oliveira, T. C. (2017b). Automated approach for asset integration in eclipse IDE. In *2017 IEEE/ACM Joint 5th Interna-*

tional Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, JSOSICSE, Buenos Aires, Argentina, May 23, 2017, pages 34–40.

[Christidis and Devetsikiotis 2016] Christidis, K. and Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303.

[France and Rumpe 2007] France, R. B. and Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. In *2007 Future of Software Engineering*, FOSE '07, pages 37–54.

[Gartner 2013] Gartner (2013). It asset management: It's all about process. Technical report, Gartner, Inc. Available at: <http://www.gartner.com/imagesrv/media-products/pdf/provance/provance_issue1.pdf>.

[K. Panetta 2018] K. Panetta (2018). 5 trends emerge in the gartner hype cycle for emerging technologies, technical report, gartner 2018. Acesso em: 10 de setembro de 2018. Disponível em <https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/>.

[Mikkonen et al. 2018] Mikkonen, T., Lassenius, C., Männistö, T., Oivo, M., and Järvinen, J. (2018). Continuous and collaborative technology transfer: Software engineering research with real-time industry impact. *Information and Software Technology*, 95:34 – 45.

[Mussbacher et al. 2014] Mussbacher, G., Amyot, D., Breu, R., Bruel, J.-M., Cheng, B. H., Collet, P., Combemale, B., France, R. B., Heldal, R., Hill, J., Kienzle, J., Schöttle, M., Steimann, F., Stikkolorum, D., and Whittle, J. (2014). The relevance of model-driven engineering thirty years from now. In *Model-Driven Engineering Languages and Systems*, pages 183–200.

[Neto et al. 2019] Neto, V. V. G., Basso, F., Santos, R., Hasrina, N., Kassab, M., Werner, C., de Oliveira, T. C., and Nakagawa, E. Y. (2019). Model-driven engineering ecosystems. In *2019 IEEE/ACM Joint 7th International Workshop on Software Engineering for Systems-of-Systems and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, JSOSICSE, Montrál, Canada, May 28, 2019*, pages 58–61.

[Paulon et al. 2014] Paulon, A., Frohlich, A., Becker, L., and Basso, F. (2014). Wireless sensor network uml profile to support model-driven development. In *12th IEEE International Conference on Industrial Informatics, At Porto Alegre, RS, Brazil*, INDIN 2014, pages 227–232.

[Ritala et al. 2014] Ritala, P., Golnam, A., and Wegmann, A. (2014). Coopetition-based business models: The case of amazon.com. *Industrial Marketing Management*, 43(2):236 – 249.

[Sommerville 2010] Sommerville, I. (2010). *Software Engineering (9th Edition)*. Addison-Wesley.

[Szabo 1996] Szabo, N. (1996). Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)*, 18.