

# Technical Aspects of Software Development in Startups: A Systematic Mapping

Bruno Henrique Cavalcante  
Programa de Pós-Graduação em  
Ciência da Computação  
Universidade Estadual de Maringá  
Maringá, Paraná

Gislaine Camila Lapasini Leal  
e Renato Balancieri  
Programa de Pós-Graduação em  
Ciência da Computação  
Universidade Estadual de Maringá  
Maringá, Paraná

Ivaldir de Farias Junior  
Softex Recife  
Recife, Pernambuco

**Abstract**—Startup companies are seeking for a business model by means of innovative products and need external investments until they can stabilize this product in the market and then start to grow to become a mature company. Nevertheless, many of these companies fail before the growth phase, emerging the need to seek for methods that help these companies to achieve their goals. This paper intends to identify the techniques, practices and tools used in software development on startups and from the results present a knowledge base that can be transferable to the industry. A systematic mapping study was conducted using a classification schema to attest the primary studies quality as well as a exclusion and inclusion criteria to select them. A total of 112 studies are found at the end of the searches and 19 of them were selected to form the mapping study. From these studies a total of 24 techniques, 31 practices and 37 tools were found. As a result the mapping study presents a technical knowledge base that seeks to fill the research gap and that can also be used as a starting point for both researchers and startups' practitioners.

**Keywords**—Software Development, Startups, Systematic Mapping Study

## I. INTRODUÇÃO

*Startups* são pequenas organizações que estão buscando por um modelo de negócio, por meio da oferta de um produto ou serviço, que possam expandir exponencialmente, uma vez que estas organizações necessitam de investimentos externos para operar [1, 2]. É comum que estas organizações acabem por criar mercados até então desconhecidos, entregando ao público produtos ou serviços que os mesmos até então sequer tinham necessidade.

Crowne [3] descreve a evolução das *startups*, da ideia até a maturidade, por meio de uma divisão em três fases, do pontapé inicial, passando pela estabilização e alcançando o crescimento. Ao fim destas fases a organização deixa de ser uma *startup* e se transforma em uma organização madura. A mesma possui um produto estável no mercado que atende as necessidades de seus consumidores e passa a ter a habilidade de desenvolver novos produtos e lançá-los no mercado. Chegar a esta fase de maturidade é, entretanto, um grande desafio para as *startups*, visto que a maioria destas organizações não conseguem atingir dois anos de vida [3]. Portanto, entender

como as *startups* se beneficiam de suas práticas de trabalho é importante para dar suporte aos novos empreendedores [1].

Conforme levantado por Paternoster et al. [1], um estudo de 1994 realizado por Carmel [4] em uma *startup*, já apontava para a necessidade de mais pesquisas quanto as práticas de desenvolvimento de software, com o intuito de replicar tais práticas e então transferi-las para outros setores da indústria de tecnologia. Porém, apesar do desenvolvimento de software ser o centro das atividades de uma *startup*, este não possui uma base de conhecimento cientificamente suportada [1].

Entre os trabalhos relacionados estão os mapeamentos desenvolvidos por Paternoster et al. [1] e Klotins et al. [5]. Paternoster et al. [1] apresentam uma análise da literatura de desenvolvimento de software em *startups* e identificam as práticas de trabalho realizadas nestas organizações. Os autores concluíram que os estudos encontrados não eram suficientes para entender o desenvolvimento de software nas *startups*, sendo que até 2014, ano do estudo, a lacuna apresentada por Sutton [2] a mais de uma década, onde este relatou a falta de estudos voltados para as *startups*, foi apenas parcialmente preenchida. Em 2015, Klotins et al. [5] buscaram aprofundar o estudo das práticas de trabalho relacionadas as *startups* por meio de um mapeamento sistemático com o objetivo de identificar e classificar as áreas de conhecimento de Engenharia de Software dentro das mesmas, porém, mais uma vez os estudos encontrados foram insuficientes, mantendo desta forma em aberto a lacuna apontada por Sutton [2].

Além disto, na agenda de pesquisa apresentada por Unterkalmsteiner et al. [6] é apontada a necessidade de estudos que verifiquem o uso de ferramentas de suporte no contexto de *startups*, visto que até o momento a investigação é insuficiente. Diante deste contexto este trabalho busca identificar, a partir de um mapeamento sistemático, as técnicas, práticas e ferramentas usadas no âmbito do desenvolvimento de software em *startups*. Identificar tais questões é importante tanto para os profissionais da área, ao fornecer por meio da Engenharia de Software Baseada em Evidência (ESBE) evidências suficientes para a transferência deste conhecimento para a indústria [7, 8],

quanto para os pesquisadores, que poderão identificar por meio do mapeamento áreas de pesquisa que ainda são pouco exploradas ou áreas que carecem de uma análise mais profunda e que se beneficiariam de uma Revisão Sistemática de Literatura (RSL).

Ao focar a pesquisa na busca por técnicas, práticas e ferramentas utilizadas no desenvolvimento de software em *startups*, este estudo tem por objetivo criar uma base de conhecimento que pode ser transferida para a indústria de forma ágil, podendo, no curto prazo, fornecer suporte para *startups* que estejam adentrando o mercado ou, até mesmo, em fase de crescimento. Ao observar quais as técnicas, práticas e ferramentas são as mais utilizadas pela indústria, estas organizações poderiam tomar decisões melhores embasadas e empiricamente validadas, guiando assim as suas escolhas.

## II. MAPEAMENTO SISTEMÁTICO

O Mapeamento Sistemático da Literatura (MSL) proporciona uma visão geral dos estudos e seus resultados, pois possibilita a identificação de lacunas, onde novos ou melhores estudos primários são necessários, ou de conjuntos de estudos que se beneficiariam de uma RSL [9]. Um MSL é recomendado para áreas de pesquisa que possuem uma falta de estudos primários relevantes e de boa qualidade [10], permitindo uma análise geral do estado de publicação na área, ajudando desta forma na tomada de decisão quanto a validade de se fazer um estudo mais profundo na área, ou se ainda é necessário que mais estudos primários sejam publicados para que isto se torne viável [9]. As etapas do MSL, de uma forma genérica, consistem em: definição da questão de pesquisa; da condução da pesquisa; triagem dos artigos; escolha de palavras-chave baseadas nos *abstracts* e, por fim, extração dos dados e mapeamento propriamente dito.

O protocolo foi avaliado por quatro especialistas e os pontos levantados foram: i) decidiu-se prosseguir com a execução das buscas manuais, até como forma de evitar que estudos potenciais não fossem encontrados devido a alguma limitação da *string* de busca eletrônica; ii) decidiu-se manter o padrão direto e instrutivo dos critérios de exclusão, deixando as questões subjetivas para serem tratadas pelos critérios de inclusão; iii) o processo de seleção visa tratar exclusivamente da reunião de todos os potenciais estudos, sendo que o produto do último passo do mesmo será a lista final de artigos selecionados. Desta forma, a avaliação da qualidade é entendida como atividade a ser executada após o processo de seleção, visto que o objetivo é avaliar a qualidade apenas dos artigos selecionados. Quanto ao *snowballing* aplica-se o mesmo princípio, visto que o objetivo é aplicar a técnica apenas no conjunto final de estudos, pois aplicá-la ainda durante a fase de seleção pode gerar desperdício de esforço em casos onde o estudo fosse eventualmente excluído antes da seleção final; iv) a princípio o protocolo busca apenas realizar esta compilação sintética das informações extraídas, porém, a partir da obtenção destes dados será conduzida uma análise das informações; v) a conferência sugerida *International Con-*

*ference on Software Engineering and Knowledge Engineering (SEKE)* foi adicionada como fonte para a busca manual.

### A. Questão de Pesquisa

O objetivo deste estudo é identificar as técnicas, práticas e ferramentas utilizadas no desenvolvimento de software em *startups*. Desta forma, a questão de pesquisa (QP1) é:

- **QP1: Quais são as técnicas, práticas e ferramentas utilizadas para o desenvolvimento de software em startups?**

### B. Estratégia de Busca

Seguindo as diretrizes sugeridas por Kitchenham and Charters [10], os seguintes pontos foram levados em consideração para a montagem da *string* de busca: i) a partir da pergunta de pesquisa são identificados os termos principais que irão compor a *string*; ii) são identificados os sinônimos dos termos; iii) todos os termos são traduzidos para o inglês, já que este é o idioma mais utilizado na literatura de Ciência da Computação, assim como em outras áreas; iv) a *string* é composta pela combinação dos termos definidos, principais e sinônimos, utilizando-se dos operadores lógicos OR (ou) e AND (e), sendo estes, respectivamente, utilizados entre os sinônimos e os termos.

Ao fim da montagem a *string* geral de busca definida é:

*software startup\* OR software start-up\* OR start-up compan\* OR startup compan\* OR lean startup\* OR lean start-up\* OR IT start-up\* OR IT startup\* OR internet start-up\* OR internet startup\* OR web startup\* OR web start-up\* OR mobile startup\* OR mobile start-up\* AND (develop\* OR engineer\* OR implement\* OR cod\* OR creat\* OR build\*)*

1) *Fontes de Busca*: Para as fontes de busca automáticas foram observadas as fontes já utilizadas por especialistas em ESBE [10] e observadas as fontes utilizadas em revisões anteriores sobre o desenvolvimento de software em *startups*, tendo como principal o trabalho realizado por Paternoster et al. [1]. As seguintes fontes foram selecionadas para as buscas eletrônicas: *Inspec/Compendex, IEEE Xplore, Scopus, ScienceDirect, ACM Digital Library e Google Scholar*. Para as fontes de busca manuais foram observadas as fontes com maior probabilidade de retornar artigos de qualidade [11], onde muitas delas são meios de publicação das fontes eletrônicas. Além disto, foram selecionadas algumas das fontes manuais verificadas por Klotins et al. [5].

### C. Estratégia de Seleção

De acordo com Kitchenham and Charters [10], a estratégia de seleção deve identificar os estudos que proporcionem uma evidência clara sobre a questão de pesquisa, desta forma, para evitar vieses, os critérios de exclusão e inclusão devem ser definidos durante a definição do protocolo.

1) *Cr terios de Exclus o*: Foram exclu dos os artigos:

- (Restri o 1) Obsoletos (mais de 10 anos, visto que um dos focos da pesquisa   permitir a transfer ncia do conhecimento para a ind stria);
- (Restri o 2) N o revisados em pares (literatura cinza);
- (Restri o 3) Que n o estejam escritos em ingl s;
- (Restri o 4) Relacionado a organiza es que n o se caracterizam como *startups*;
- (Restri o 5) N o relacionado ao processo de desenvolvimento de software.
- (Restri o 6) Que contemplam somente a execu o de estudos te ricos sobre *startups*.
- (Restri o 7) Que n o estejam dispon veis na internet;
- (Restri o 8) Que n o consigam responder satisfatoriamente a pergunta de pesquisa;
- (Restri o 9) Quando mais de um artigo conter os mesmos resultados publicados, os menos detalhados ser o exclu dos;
- (Restri o 10) Quando forem encontrados artigos duplicados, apenas o mais completo ser  mantido.

2) *Cr terios de Inclus o*: Foram inclu dos os artigos:

- (Restri o 11) Que contemplem a execu o de estudos emp ricos a respeito das t cnicas, pr ticas e ferramentas utilizadas no desenvolvimento de software dentro do contexto de *startups*.
- (Restri o 12) Quando mais de um artigo conter resultados distintos de um mesmo estudo, todos ser o inclu dos.
- (Restri o 13) Durante as buscas manuais artigos publicados em portugu s tamb m ser o aceitos, abrindo assim uma exce o para a Restri o 3.

3) *Processo de Sele o*: O processo de sele o dos estudos foi conduzido a partir de quatro etapas.

a) *Aplica o da String em Bases Eletr nicas*: Neste passo foram realizadas as buscas nas bases eletr nicas a fim de encontrar os estudos prim rios. Foram lidos o t tulo, *abstract* e as palavras-chaves de cada estudo observando-os em rela o aos cr terios de exclus o definidos anteriormente. Aqueles artigos que n o geraram confian a suficiente para serem exclu dos foram mantidos para an lise em um passo futuro.

As fontes de busca eletr nica juntamente com a quantidade total de artigos retornados e de artigos aceitos durante a aplica o da *string* podem ser visualizadas em <https://tinyurl.com/ybdlnalw>.

Ao fim da busca, foram iniciadas as leituras de t tulo, *abstract* e palavras-chave dos artigos retornados, observando-os em rela o aos cr terios de inclus o e exclus o definidos no protocolo.

b) *Busca Manual*: Neste passo foram realizadas as buscas nas fontes manuais onde jornais e revistas foram analisados ao percorrer a lista de artigos publicados nos volumes e edi es da publica o. J  para as confer ncias a busca foi feita ao analisar os artigos aceitos para publica o. As fontes

utilizadas (revistas, jornais e confer ncias) j  com a quantidade de artigos retornados por cada fonte podem ser visualizadas em <https://tinyurl.com/y75oae9u>.

Foram lidos o t tulo, *abstract* e as palavras-chaves de cada estudo observando-os em rela o aos cr terios de exclus o e inclus o definidos anteriormente. Aqueles artigos que n o geraram confian a suficiente para serem exclu dos foram mantidos para an lise em um passo futuro. Durante este passo,   comum que uma grande quantidade de artigos totalmente irrelevantes para a pergunta de pesquisa sejam encontrados [10], ou seja, que n o est o relacionados ao desenvolvimento de software ou a *startups*, desta forma, estes artigos foram permanentemente descartados, n o sendo registrados nas listas de artigos exclu dos. Como forma de complementar a busca manual, foi utilizada a t cnica de *snowballing*, que consiste em analisar os artigos referenciados nos estudos encontrados e avaliar se os mesmos devem ser inclusos no conjunto final ou n o. A t cnica foi conduzida durante a etapa de extra o de dados dos artigos selecionados.

c) *Uni o de Listas*: Neste passo foi realizada a uni o das listas de artigos encontrados durante as buscas. A uni o das listas dos artigos encontrados durante as buscas eletr nicas (59) e manuais (71) resultou em um total de 130 artigos encontrados, por m destes, 18 artigos encontrados na busca manual foram considerados duplicados pois j  haviam sido retornados durante a busca eletr nica, sendo ent o removidos das futuras listas.

d) *Sele o Final*: Por fim, neste passo realizou-se a leitura por completo dos potenciais estudos obtidos do passo anterior e, tamb m,   nesta etapa que os artigos que geraram incertezas nos passos anteriores foram novamente analisados. Durante a an lise dos estudos foram observados os cr terios de exclus o e os cr terios de inclus o, onde ao fim desta an lise obteve-se a sele o final de artigos. Ap s a uni o das listas um total 112 artigos chegaram at  esta fase para serem analisados. Ao t rmino da leitura, um total de 17 artigos foram aceitos por satisfazerem a pergunta de pesquisa proposta no protocolo e ent o obteve-se a sele o final de artigos e considerou-se finalizada a etapa de sele o dos estudos. A Fig. 1 sintetiza as etapas e resultados do processo de sele o.

A Fig. 2 demonstra a distribui o por ano onde, por meio da linha de tend ncia,   poss vel notar a evolu o do tema ao longo dos anos.

4) *Avalia o da Qualidade*: Segundo Kitchenham and Charters [10], a qualidade de um estudo est  relacionado com a sua habilidade de minimizar o vi s e maximizar a valida o interna e externa. Um estudo enviesado   aquele onde os resultados se afastam dos resultados reais, uma vez que um artigo imparcial, ao contr rio de enviesado, ser  internamente v lido. Desta forma, um estudo internamente v lido   aquele onde os resultados s o os mais pr ximos poss veis da realidade, sendo este um requisito para a valida o externa. A valida o externa diz respeito a replica o do estudo, ou seja, se os efeitos observados no estudo podem ser alcan ados fora dele.

Avaliar a qualidade dos estudos permite verificar a import ncia de estudos individuais durante a sintetiza o dos

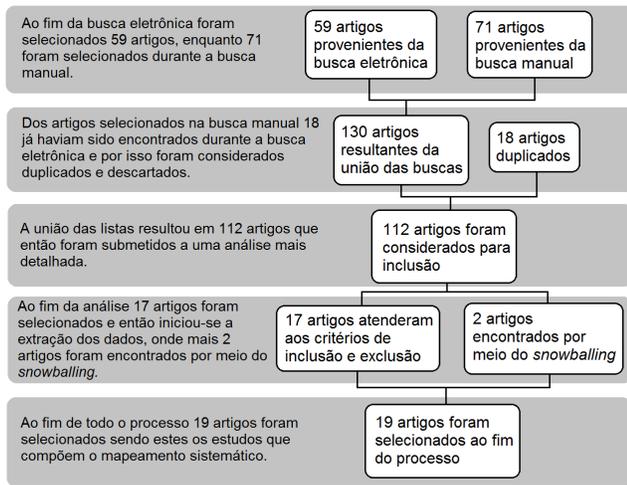


Fig. 1. Resultados do processo de seleção.

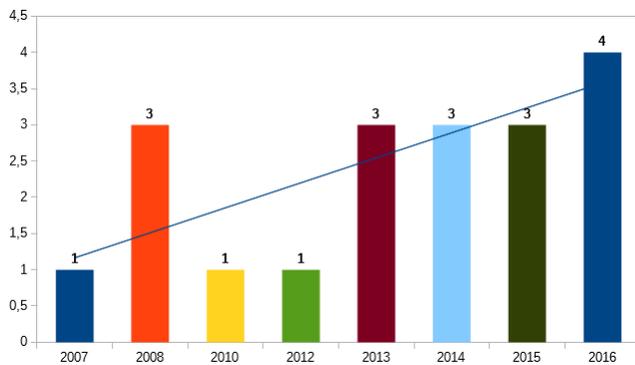


Fig. 2. Distribuição dos artigos aceitos por ano durante a seleção final.

resultados, auxilia a guiar as recomendações para pesquisas futuras e também a guiar a interpretação dos resultados e determinar a força das inferências [12].

Então, para avaliação da qualidade dos estudos selecionados os mesmos foram verificados em relação as questões abaixo por meio de respostas de "Sim" ou "Não" [13]:

- 1) O artigo é baseado em pesquisas ou é apenas um relatório de lições aprendidas com base na opinião de especialistas?
- 2) Há uma declaração clara dos objetivos da pesquisa?
- 3) Existe uma descrição adequada do contexto no qual a pesquisa foi realizada?
- 4) O projeto da pesquisa foi adequado para resolver os objetivos da pesquisa?
- 5) A estratégia de recrutamento foi adequada aos objetivos da pesquisa?
- 6) Havia um grupo de controle com o qual se comparar os tratamentos?
- 7) Os dados foram coletados de uma forma que abordou a questão de pesquisa?
- 8) A análise de dados foi suficientemente rigorosa?
- 9) A relação entre o pesquisador e os participantes foi

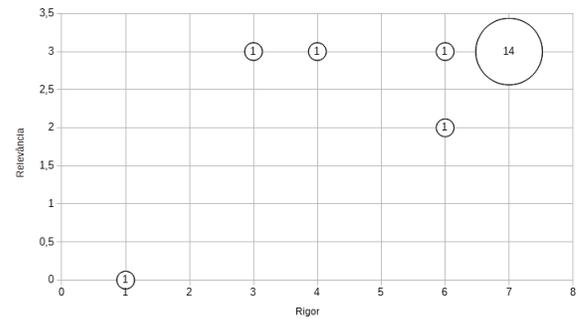


Fig. 3. Avaliação dos critérios de rigor e relevância.

devidamente considerada?

10) Há uma declaração clara dos resultados?

11) O estudo é de valor para a pesquisa ou a prática?

Como forma de medir a qualidade dos estudos as questões foram particionadas sob os critérios de rigor e relevância. O rigor representa a precisão utilizada pelo estudo em seu método de pesquisa e a forma como o estudo é apresentado. A relevância representa o valor do estudo para a comunidade de pesquisa e para a indústria [14]. As questões 1 a 8 estão relacionadas ao rigor e as questões 9 a 11 são referentes a relevância. Cada resposta "Sim" possui valor 1 e cada resposta "Não" possui valor 0, sendo assim, o valor máximo para o critério de rigor é 8 e o valor máximo para o critério de relevância é 3.

A Fig. 3 demonstra a visão geral dos artigos encontrados analisados por meio dos critérios de rigor e relevância. Nota-se que a maior parte dos artigos (16) estão localizados no canto superior direito do gráfico, o que denota alto rigor e alta relevância. Apenas 2 artigos possuem um rigor mediano, porém ainda apresentam máxima relevância. Por fim, apenas 1 dos artigos apresenta baixo rigor e baixa relevância.

Além das questões, também foi preenchido um esquema de classificação para listagem final de artigos. O esquema de classificação é adaptado de um MSL sobre desenvolvimento de software em *startups* [1], e com ele busca-se categorizar os estudos encontrados de uma forma que seja possível determinar sua qualidade. O esquema é particionado sob 4 aspectos: tipo de pesquisa, tipo de contribuição, foco e significância.

O tipo de pesquisa visa identificar os diferentes tipos de estudo (pesquisa de avaliação, proposta de solução, artigo filosófico, artigo de opinião ou artigo de experiência), sem considerar as metodologias empregadas em cada estudo.

O tipo de contribuição permite identificar o tipo de contribuição que o estudo proporciona, dos quais estes tipos podem ser divididos entre fracos (diretrizes, lições aprendidas, conselho/implicações e ferramenta) e fortes (modelo, teoria e *framework/métodos*).

Em relação ao foco, suas categorias (desenvolvimento de software, gestão de processo, ferramentas e tecnologias, gestão/organizacional) permitem identificar os estudos diretamente relacionados ao desenvolvimento de software, as

ferramentas e tecnologias utilizadas para suportar o desenvolvimento ou então aqueles que possuem maior foco quanto a gestão do processo de desenvolvimento.

Quanto ao aspecto de significância (total, parcial e marginal), o mesmo permite identificar o grau de relacionamento com as atividades de engenharia em *startups*, onde aqueles estudos com maior relação terão maior significância para a pergunta de pesquisa.

Classificar os estudos sob estes aspectos permite verificar a relevância dos estudos em relação a questão de pesquisa. Ao analisar a significância dos estudos, por exemplo, pode-se verificar quais estudos estão diretamente ligados ao desenvolvimento de software em *startups* e por consequência quais serão aqueles que provavelmente terão maiores contribuições para responder a questão de pesquisa. De maneira geral, os aspectos permitem identificar de uma forma ágil os tipos de resultados que são esperados de cada estudo e assim guiar o leitor para aqueles estudos que provavelmente serão mais relevantes para o mesmo.

#### D. Estratégia de Extração

Para Kitchenham and Charters [10] a extração dos dados deve ser projetada para coletar toda a informação necessária para responder a questão de pesquisa, assim como para analisar os estudos por meio dos critérios de seleção. Desta forma, seguindo as diretrizes [10], as seguintes informações devem ser extraídas a fim de se formar um padrão: i) Título do artigo; ii) Autor(es); iii) Ano de publicação; iv) Tipo de fonte (revista, conferência ou jornal); v) Título da fonte. Em adição, seguindo o esquema de classificação descrito na seção anterior, os seguintes dados foram extraídos de acordo com os aspectos contidos no esquema: i) Tipo de pesquisa; ii) Tipo de contribuição; iii) Foco; iv) Significância.

Durante esta etapa de extração dos dados foi aplicada a técnica de *snowballing* para os estudos selecionados, dos quais aqueles estudos que se mostraram promissores foram analisados por meio dos critérios de inclusão e exclusão. A técnica consiste em verificar a lista dos artigos referenciados nos estudos aceitos com intuito de identificar novos artigos para inclusão [15]. Os artigos que se mostraram irrelevantes por meio do título ou foram considerados obsoletos não entraram na lista de artigos excluídos, sendo permanentemente descartados. Durante a aplicação da técnica um total de 21 potenciais artigos foram encontrados, porém, 17 deles foram considerados duplicados, restando apenas 4 artigos que foram analisados, e após a análise dos mesmos apenas 2 foram aceitos observando todos os critérios de inclusão e exclusão. Ao fim do *snowballing* um total de 19 artigos foram aceitos durante toda a busca, sendo estes os artigos que tiveram os dados extraídos.

A Fig. 4 demonstra a distribuição e relacionamento dos artigos encontrados por meio dos aspectos tipo de pesquisa, tipo de contribuição e foco. Nesta figura pode-se verificar que o cruzamento entre o tipo de contribuição lições aprendidas e o tipo de foco desenvolvimento de software representam o maior número de artigos (6). Ao cruzar o tipo de pesquisa com o

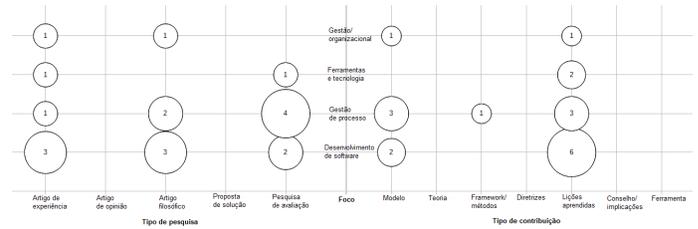


Fig. 4. Tipo de pesquisa, tipo de contribuição e foco.

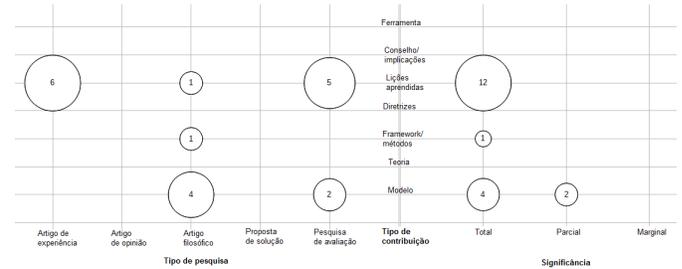


Fig. 5. Tipo de pesquisa, significância e tipo de contribuição.

foco, o maior número de artigos (4) está entre as categorias de pesquisa de avaliação e gestão de processo. Separadamente, o tipo de contribuição lições aprendidas agrega o maior número de artigos, com 12. Por sua vez, o tipo de pesquisa de avaliação representa 7 artigos, enquanto para o foco a categoria desenvolvimento de software, com 8 artigos, representa o maior número.

A Fig. 5 apresenta o relacionamento entre os aspectos tipo de pesquisa, significância e tipo de contribuição. Nesta figura nota-se que os cruzamentos entre o foco lições aprendidas e significância total representam um total de 12 artigos, enquanto ao cruzar o foco lições aprendidas com o tipo de pesquisa artigo de experiência temos um total de 6 artigos. Esta figura também demonstra que apenas 2 artigos não possuem total relação com as atividades de engenharia de software em *startups*.

Por fim, a Fig. 6 demonstra a relação entre os aspectos tipo de pesquisa, significância e foco. A figura demonstra que o cruzamento entre a significância total e tipo de pesquisa pesquisa de avaliação (7) e significância total de foco desenvolvimento de software (8) são onde estão concentrados a maioria dos artigos.

A extração dos dados dos 19 artigos encontrados no mapeamento

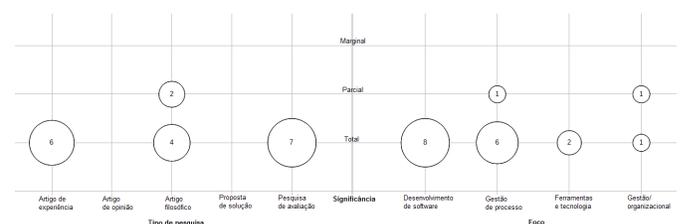


Fig. 6. Tipo de pesquisa, significância e foco.

mento pode ser visualizada em <https://tinyurl.com/yacbq84s>, onde são apresentados os títulos dos artigos e seus respectivos autores, ano de publicação, tipo e título da fonte na qual os mesmos foram publicados.

### III. RESULTADOS E DISCUSSÃO

Nesta seção são apresentadas as técnicas, práticas e ferramentas extraídas, visando proporcionar ao profissional da indústria uma visão sobre a utilização das mesmas e assim apoiar o mesmo nos processos de decisão da *startup*. Também são discutidos os achados de acordo com o esquema de classificação e as áreas de pesquisa que são pouco exploradas e que se beneficiariam de uma análise mais detalhada. A seção está particionada entre as contribuições para a indústria e as contribuições para a academia.

#### A. Contribuição para a Indústria

Como contribuição para a indústria são citadas as técnicas, práticas e ferramentas encontradas durante a extração. Nesta subseção é apresentada uma breve descrição e que tipo de contribuição cada um dos estudos encontrados demonstra, de forma que o leitor possa identificar aqueles artigos que mais lhe interessam e então se aprofundar nos mesmos.

Dos 19 artigos encontrados foram extraídas 24 técnicas, 31 práticas e 37 ferramentas. Dentre as técnicas extraídas, a que apresenta maior frequência entre os estudos é a metodologia *Lean/Lean Startup* com 7 ocorrências, seguida pela metodologia Scrum com 4. Metodologias ágeis em geral, Kanban e cartões de estórias aparecem 3 vezes nos artigos encontrados, seguidas por *Extreme Programming*, *Planning Game*, *Backlog* e programação em pares com duas ocorrências. As demais técnicas encontradas (*Join Application Design*, *Experiment Driven Development* (EDD), *Teste A/B*, *Feature Driven Development* (FDD), *Rational Unified Process* (RUP), *Scrumban*, *logs* e estatísticas e *Code Review*) aparecem apenas uma vez cada. Técnicas de *Minimum Viable Product* (MVP) aparecem 7 vezes distribuídas em tipos diferentes.

Em relação as práticas, integração contínua possui a maior frequência, aparecendo 5 vezes entre os artigos. Utilização de soluções *commercial off-the-shelf* (COTS), conhecido como software de “prateleira”, ou *open source*, padrões de código/*design/frameworks*, refatoração de código e entregas frequentes/contínua são as próximas com 4 ocorrências cada.

Testes de aceitação e reuso de software contam com 3 ocorrências, seguidas por cliente no local de desenvolvimento e testes de integração com duas ocorrências. O restante das práticas (semana de trabalho de 40 horas, código pertence a todos, teste contínuo, *design* simples, *releases* curtos, teste unitário, teste de usabilidade, entrega manual, gerenciamento de requisitos, apresentação de materiais, backups diários em produção, *rollback* automático em produção em caso de falha, *bug-tracking*, capacitação dos membros da equipe, indicadores chave de performance, prototipagem evolucionária, controle de versão, entrevistas, usuários de testes, pesquisa de mercado, comando e controle e incluir e capacitar) possuem uma ocorrência cada.

No conjunto das ferramentas extraídas a que possui maior ocorrência é o Google Analytics, com apenas duas, onde o restante das ferramentas (Atlassian Jira, Axosoft OnTime, Target Process, Microsoft TFS, Rally Platform, Mingle, Version One, Blossom.io, Scrumwise, Base Camp, LeanKit, AgileZen, PlanBox, Kanbanize, ScrumWorksPro, BananaScrum, AgileFant, IceScrum, Xplanner, Trello, Asana, XPStoryStudio, JustInMind, Betalist.com, Mixpanel, Hadoop, Storm, Kafka, Flume, Hbase e AIMLBot) e linguagens (Haskell, C++, PHP, Python e .NET) aparecem apenas uma vez cada.

Dos artigos considerados para a seleção final destaca-se o mapeamento sistemático de literatura realizado por Paternoster et al. [1]. Apesar de abrangente, uma das perguntas de pesquisa do estudo foca nas práticas de trabalho relacionadas a engenharia de software em *startups*, sendo que os achados desta pergunta proporcionaram um volume considerável de técnicas e práticas extraídas com 20 contribuições. O trabalho de Tingling and Saeed [16] apresentaram uma contribuição considerável, com 11 ocorrências entre técnicas e práticas. No estudo, os autores identificam e analisam a adoção dos princípios de XP em uma *startup*, onde, curiosamente, a programação em pares não é adotada, mesmo sendo uma das principais técnicas da metodologia.

O estudo desenvolvido por Martin et al. [17] é focado em testes e faz uma análise de como os testes são conduzidos em uma *startup*. Várias das práticas de trabalho identificadas na *startup* são derivadas da metodologia XP, entretanto não se pode dizer que a organização de fato segue a metodologia já que, assim como no trabalho de Tingling and Saeed [16], a programação em pares não é adotada. No caso de estudo analisado por Martin et al. [17] foi identificado que os processos de testes ainda precisam ser pesquisados mais a fundo, pois ainda há uma distância entre o que se tem em pesquisas e a realidade. Outro estudo focado em uma das partes que compõem o desenvolvimento, desta vez voltado a interface, mais especificamente *User Experience* (UX), é apresentado por Hokkanen and Väänänen-Vainio-Mattila [18]. O estudo investiga, por meio de entrevistas, as práticas atuais relacionadas a UX dentro de *startups*. Ao fim do estudo são apontados alguns passos a serem seguidos por *startups*, visto que de acordo com os autores os processos de UX são de grande importância para estas organizações, principalmente para aquelas que estão planejando entrar em novos mercados por meio de produtos inovadores. Em outro estudo, Duc and Abrahamsson [19] conduzem uma investigação a respeito do papel do MVP em *startups*, do qual são apresentados alguns dos principais tipos de MVP utilizados neste contexto. O estudo observou que as técnicas de MVP são úteis pois os artefatos gerados servem como limites do que pode ser realizado e também podem ser reutilizados em outros estágios do projeto. Além disto, por meio dos estudos de caso, o estudo identificou que metodologias de desenvolvimento ágil são o processo mais viável para *startups*, pois permitem entregas rápidas e encurtam o tempo entre a concepção da ideia e a produção da mesma.

Na pesquisa realizada por Langer and Vaněček [20] é con-

duzida uma investigação breve a respeito de qual metodologia será utilizada para um determinado projeto de uma *startup*, sendo apontada ao final a metodologia considerada mais adequada, no caso FDD, pois a mesma proporcionou vantagens como a possibilidade de mudanças no escopo e uma lista das funções que podem ser facilmente implementadas no futuro. Lindgren and Münch [21] por sua vez apresentam um estudo investigativo a respeito do estado da experimentação no desenvolvimento de produto. Apesar de não ser focado especificamente em *startups* alguns dos casos de estudo analisados podem ser consideradas como *startups*, onde é possível identificar as metodologias usadas nas mesmas, além de que, em pelo menos uma, utiliza-se da experimentação no desenvolvimento do produto.

Giardino et al. [22] conduziram um estudo visando criar-se um modelo de auxílio às *startups* para que estas possam lançar seus produtos no mercado com maior agilidade. Durante as investigações é possível identificar algumas práticas utilizadas em *startups*, sendo estas informações principalmente de fontes externas. Este estudo é realizado por meio de uma RSL e, assim como os estudos realizados por Paternoster et al. [1] e Klotins et al. [5], aponta que existe uma falta de pesquisas que suportem as atividades de desenvolvimento de software em *startups*. Outro estudo de Giardino et al. [23] questiona o que se sabe sobre o desenvolvimento de software em *startups*, apresentando algumas das técnicas e práticas que são utilizadas no contexto, além de trazer pontos interessantes sobre quais seriam os fatores responsáveis pelo sucesso de uma *startup*. Coleman and O'Connor [24] buscam identificar como é formado o processo de desenvolvimento de software dentro de *startups*, tendo-se como principal influência as experiências anteriores daqueles que são os encarregados do trabalho de desenvolvimento. No estudo é possível identificar algumas metodologias e formas de gerenciamento que são utilizadas. Entre os achados do estudo, a abordagem de gerenciamento "incluir e capacitar" se mostra interessante pois proporciona liberdade ao time de desenvolvimento, que necessita de menos supervisão direta, seguindo na mesma direção que um dos pontos apontados por Giardino et al. [23] como sendo fator de sucesso para *startups*.

Alguns trabalhos a respeito da metodologia *Lean/Lean Startup* foram encontrados. Miski [25] apresenta um estudo acerca do desenvolvimento de uma aplicação *mobile* utilizando da metodologia *Lean Startup*. Taipale [26] apresenta um estudo parecido, porém de forma mais abrangente, não focando no desenvolvimento de apenas uma aplicação. Björk et al. [27] propõem um modelo de auxílio para que *startups* possam aplicar os princípios do *Lean Startup*, visto que o estudo aponta que a metodologia é pouco aplicada pois seus métodos são muito vagos e imprecisos para serem implementados na prática. Das organizações investigadas durante as entrevistas todas utilizam algum tipo de metodologia ágil, principalmente Scrum e Kanban. Škrabálek and Böhm [28] buscam criar uma abordagem *Lean* por meio de um caso de estudo em uma *startup*, do qual são identificadas algumas das práticas utilizadas pela mesma, além da metodologia Scrum. No estudo

de Terho et al. [29] duas *startups* são investigadas em relação a utilização do *Lean Startup*, sendo que estas fazem uso de MVP para iterar por meio de um processo de construção, medição e aprendizagem, o que de acordo com os achados aumenta a produtividade e reduz o *time-to-market* (o tempo entre a concepção do produto até a sua disponibilização para venda).

Jansen et al. [30] conduziram um estudo sobre reuso em *startups*, onde nos casos de estudo apresentados também é possível identificar algumas das ferramentas utilizadas. O reuso se provou rentável para as *startups* analisadas no estudo visto que possibilitou a elas desenvolverem seus produtos por meio da reutilização de funcionalidades que as mesmas não conseguiriam desenvolver por si próprias. Nanavati [31] apresenta as motivações e os benefícios da utilização de Haskell como principal linguagem de programação em uma *startup*. O uso da linguagem se provou como um forte diferencial para a *startup* em foco no estudo e, segundo o autor, linguagens de programação puras como o Haskell podem ser fonte de uma vantagem competitiva para *startups* visto que as vantagens trazidas pela pureza são muito maiores que as desvantagens. Em outro artigo, apesar de não ser o foco do estudo de De Moura et al. [32], o trabalho apresenta as linguagens de programação e ferramentas utilizadas pela *startup* investigada no estudo. Por fim, Taheri and Sadjadi [33] montam um esquema de seleção de ferramentas baseado em característica para o desenvolvimento de software ágil. Apesar de não ser focado apenas em *startups*, o esquema gerado apresenta uma distribuição de ferramentas de acordo com uma classificação para o uso das mesmas nesta fase de vida de uma organização.

## B. Contribuição para a Academia

Nesta subseção são discutidos os principais pontos investigados pelos estudos e é destacada uma visão geral das contribuições dos mesmos. Além disto, é apresentado um panorama dos estudos encontrados de acordo com o esquema de classificação e, também, são apontados quais tipos de estudos poderiam ser conduzidos visando preencher as lacunas da área.

Conforme demonstram os dados extraídos a partir do esquema de classificação, dos 19 artigos encontrados apenas 2 não possuem total ligação com as atividades de desenvolvimento de software em *startups*, porém ainda estão parcialmente relacionados, o que indica que os achados possuem um grande valor para este tipo de organização. Entre os artigos, 6 deles são identificados como do tipo de pesquisa artigo filosófico, ou seja, estes artigos buscam estruturar o campo de conhecimento em forma de uma taxonomia ou de um *framework* conceitual.

Outros 7 artigos são categorizados como de pesquisa de avaliação, onde uma metodologia é aplicada na prática e uma avaliação da mesma é conduzida, sendo que junto com os 6 artigos de experiência, que demonstram como algo foi feito na prática, são os que provavelmente irão fornecer um conhecimento que seja de fácil aproveitamento pela indústria. Somado a isto estão 12 artigos cujo tipo de contribuição são de lições aprendidas onde é possível analisar os resultados diretamente

obtidos em pesquisa. Ainda sobre o tipo de contribuição, apenas 1 artigo consta na categoria de *framework*/métodos, que apresenta modelos relacionados à construção de software ou gestão do processo de desenvolvimento. Os 6 artigos restantes apresentam modelos, ou seja, representam uma realidade observada por meio de conceitos.

Em relação ao aspecto foco, 8 dos artigos encontrados podem ser categorizados como de desenvolvimento de software, o que ajuda a suportar a ideia de que os estudos identificados possuem grande relevância para a indústria, visto que os artigos desta categoria estão diretamente ligados as atividades de engenharia usadas para escrever e manter o código fonte. Além disto, 7 artigos são classificados como de gestão de processo, sendo que estes estão relacionados aos métodos e técnicas utilizadas para o gerenciamento das atividades de desenvolvimento. Entretanto, apenas 2 artigos tratam especificamente dos instrumentos usados para criar, depurar, manter e suportar as atividades de desenvolvimento. Os 2 artigos restantes são categorizados como gestão/organizacional e tratam dos aspectos voltados ao gerenciamento da estrutura.

Pode-se verificar que boa parte das técnicas extraídas estão alinhadas com a agilidade necessária às *startups*. Lean e Scrum como metodologias ágeis, Lean e MVP como forma de aprender por meio de interações com o usuário são alguns dos exemplos. Em relação as práticas o mesmo pode ser percebido. Testes e ciclos curtos são algumas das práticas que mais aparecem, porém de forma muito distribuída. O mesmo pode ser dito para questões voltadas a capacitação do time. Para Giardino et al. [23], o empoderamento da equipe possui um papel importante para o sucesso de uma *startup*, visto que é o conhecimento e a capacidade da equipe que irão compensar a falta de recursos da organização. Em relação as ferramentas, são poucos os artigos que focam especificamente em estudá-las, sendo citadas esporadicamente durante os trabalhos. Apenas o estudo desenvolvido por Taheri and Sadjadi [33] é focado em analisar ferramentas, porém seu objetivo é montar um esquema de classificação para seleção de ferramentas para o desenvolvimento ágil, sem focar especificamente em *startups*.

Conforme demonstrado pelos estudos, existe uma certa diversidade entre os trabalhos a respeito do desenvolvimento de software em *startups*. O esquema de classificação gerado na extração dos dados demonstra que boa parte dos artigos encontrados são de lições aprendidas, enquanto o restante, com algumas exceções, são de modelos. São poucos os estudos que focam em determinada área, como testes ou interface, sendo que mais estudos deste tipo seriam importantes, pois ajudam na construção de uma base de conhecimento para a solução de problemas específicos. Alguns dos estudos encontrados demonstram resultados interessantes para a sobrevivência das *startups*, como por exemplo a utilização de técnicas de MVP, que permitem que estas organizações aprendam durante o desenvolvimento do produto e então consigam lançar no mercado um produto com mais chances de sucesso. A utilização de metodologias ágeis é o ponto mais comum entre os estudos, sendo que mesmo aqueles que não citam diretamente alguma metodologia ágil pelo menos faz uso de algum tipo de princí-

pio ágil. Estas metodologias são de grande valor para *startups* pois tempo é essencial para estas organizações, visto que a maioria delas falham dentro de um período curto de tempo, e as mesmas permitem um desenvolvimento mais rápido do produto por meio de entregas frequentes, maior produtividade e redução to *time-to-market*.

Estudos mais abrangentes auxiliam na visão geral do desenvolvimento dentro do contexto de *startups*, permitindo que se veja as áreas carentes de informações e que se beneficiariam de mais estudos focados. São poucos os estudos, por exemplo, que focam em avaliar ferramentas, sendo que apenas um dos estudos buscou criar um modelo para seleção de ferramentas, porém, o estudo em questão não é focado apenas em *startups*. Desta forma, estudos de avaliação ou de comparação de ferramentas são necessários. O uso de *frameworks* está entre as práticas encontradas, entretando nenhum estudo comparativo entre *frameworks* de desenvolvimento disponíveis foi identificado. Na realidade, muitas das práticas identificadas podem ser aplicadas por meio de ferramentas, porém estudos que avaliem ou comparem estas ferramentas são escassos. Portanto, estas seriam as áreas mais carentes no momento, e estudos em torno da mesma seriam de grande importância em virtude da facilidade de transferir este conhecimento para a indústria. De maneira resumida, o número de estudos ainda é pequeno, e mais trabalhos neste contexto são necessários para a formação de uma base de conhecimento sólida que possa ser transferida para indústria com maior frequência e aproveitamento.

#### IV. LIMITAÇÕES E AMEAÇAS A VALIDADE

Assim como revisões sistemáticas, a tendência a publicação de apenas bons resultados, é um possível viés ao mapeamento sistemático [10]. Entretanto, esta não é uma grande ameaça à validade do estudo conduzido visto que não é o objetivo do mesmo avaliar o desempenho, eficácia, produtividade ou qualquer outro medidor de qualidade das técnicas, práticas e ferramentas extraídas. Ainda assim, isto pode ser uma limitação já que possivelmente diminui o número de estudos publicados a respeito de *startups*, já que há uma grande tendência de que as mesmas falhem [3].

A construção da *string* de busca eletrônica é um dos principais desafios no planejamento do mapeamento sistemático. No caso das *startups* existem vários sinônimos que podem ser utilizados e com isto é preciso cuidado para não tornar a *string* muita genérica, trazendo uma quantidade quase imensurável de artigos. Como aponta Paternoster et al. [1], os termos *startup* e *start-up* se assemelham com o verbo especial *to start up* da lingua inglesa, que para outras áreas da literatura está ligada com o processo de início de movimento de um motor. Durante a busca eletrônica, vários dos artigos retornados que foram excluídos já durante a leitura de seus títulos tratavam a respeito de sistemas de controle ou segurança de plantas nucleares, onde o termo *to start up* é bastante utilizado para determinar o mesmo onde estes sistemas entram em ação. Para atenuar esta ameaça a formulação da *string* foi baseada na construída por Paternoster et al. [1], que se provou confiável durante o seu trabalho de mapeamento.

Ameaças quanto a seleção dos estudos são mitigadas por meio dos critérios de inclusão e de exclusão [10]. Porém, visto que o processo de seleção dos estudos foi realizado por apenas um pesquisador, existe a possibilidade de enviesamento na seleção em virtude de opiniões pessoais. Para tratar esta ameaça o processo de seleção foi tratado com o máximo de critério, neutralidade e imparcialidade, dedicando-se um tempo de leitura e reflexão acima da média para aqueles estudos que geraram mais incertezas. Além disto, as discussões com outros pesquisadores eram frequentes, sendo que em mais de um caso outro pesquisador foi consultado como forma de obter uma opinião a respeito da inclusão ou exclusão de algum dos estudos encontrados.

## V. CONCLUSÕES

Giardino et al. [23] apontam que para ter sucesso, *startups* precisam de métodos transferíveis e confiáveis, ou seja, que são fáceis de colocar em prática e que tragram resultados rápidos. A natureza de incerteza que compõe uma *startup* exige que os processos sejam desenvolvidos com velocidade, assim, as atividades devem ser desenhadas para permitir flexibilidade e reatividade. Inclusive, diante deste contexto, é preferível que se falhe rápido e conseqüentemente se aprenda rápido. A partir disto, Giardino et al. [23] citam alguns pontos que devem ser atendidos para permitir esta velocidade, e conseqüentemente o sucesso, às *startups*: i) uso de *frameworks* conhecidos, permitindo mudanças rápidas no produto para atender ao mercado; ii) uso de prototipagem evolucionária e experimentação por meio de componentes existentes; iii) validação contínua com grupos de usuários chave; iv) entrega de valor contínua focando em funcionalidades chaves que engajem o usuário; v) empoderamento do time como forma de aumentar o desempenho e o sucesso; vi) utilização de métricas para aprender por meio dos usuários; vii) uso de ferramentas fáceis de implementar para tornar o desenvolvimento do produto mais dinâmico.

O número de estudos no âmbito do desenvolvimento de software em *startups* que apresentam resultados transferíveis para a indústria ainda é baixo. Dentre os itens levantados por Giardino et al. [23], todos apresentam lacunas e se beneficiariam de mais estudos empíricos. São necessários estudos que busquem soluções para atender a estes itens, a fim de criar cenários onde técnicas, práticas e ferramentas possam ser validadas e, possivelmente a partir dos dados levantados, criarem-se guias de métodos confiáveis que podem ser adotados. Há também a necessidade de estudos comparativos, que busquem por exemplo avaliar os *frameworks* de desenvolvimento mais populares do mercado, e verificar sua validade para a aplicação no contexto de uma *startup*. O mesmo pode ser dito para os demais pontos levantados por Giardino et al. [23], mas principalmente quanto a questão da capacitação da equipe, já que este é um dos pontos menos explorados entre os citados.

Neste sentido, este trabalho buscou mapear as técnicas, práticas e ferramentas utilizadas no desenvolvimento de software em *startups* e assim criar uma base de conhecimento que pode servir como guia para a tomada de decisão de *startups*.

Ao identificar um total de 24 técnicas, 31 práticas e 37 ferramentas, pode-se dizer que o objetivo foi alcançado. Apesar da necessidade de um número maior de estudos empíricos para que a área de pesquisa esteja suficientemente madura, as informações coletadas neste mapeamento constituem um ponto de partida para os profissionais de *startups*.

Em comparação aos mapeamentos apresentados por Klotins et al. [5] e Paternoster et al. [1], o primeiro analisa as áreas de conhecimento de Engenharia de Software que são usadas em *startups* enquanto o segundo é mais focado nos aspectos organizacionais que compõem o desenvolvimento de software em *startups*, sendo que quando descreve a respeito das práticas encontradas o faz por meio de um panorama geral, já este mapeamento busca preencher a lacuna apontada por Sutton [2] ao focar no aspecto técnico, identificando as técnicas, práticas e ferramentas que dão suporte ao desenvolvimento de software nas *startups*, além disto, por considerar apenas estudos realizados nos últimos dez anos, espera-se que os achados do mapeamento tenham maior relevância para o cenário atual das *startups*.

Como trabalhos futuros, há a possibilidade explorar os pontos apontados por Giardino et al. [23], dos quais estes, de acordo com o autor, são os principais fatores para o sucesso de uma *startup*. Entre estes pontos, explorar os *frameworks* e ferramentas utilizados no desenvolvimento, seja por meio de avaliações ou de estudos comparativos, são áreas com grande potencial para pesquisa pois, conforme verificado por meio do mapeamento, são áreas ainda pouco exploradas e com alta possibilidade de transferência para a indústria. O mesmo pode ser dito a respeito do uso de métricas para aprender por meio dos usuários. Um forte indicador para isto é que atividades relacionadas ao uso de métricas e aprendizagem por meio do usuário aparecem em 15 dos 19 estudos em relação as técnicas extraídas, além de 4 ocorrências entre as práticas e 2 ocorrências entre as ferramentas, demonstrando que esta atividade é ponto chave no desenvolvimento de software em *startups*.

## REFERÊNCIAS

- [1] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: A systematic mapping study," *Information and Software Technology*, vol. 56, no. 10, pp. 1200–1218, 2014.
- [2] S. M. Sutton, "The role of process in software start-up," *IEEE Software*, vol. 17, no. 4, pp. 33–39, 2000.
- [3] M. Crowne, "Why software product startups fail and what to do about it. evolution of software product development in startup companies," in *Engineering Management Conference, 2002. IEMC'02. 2002 IEEE International*, vol. 1. IEEE, 2002, pp. 338–343.
- [4] E. Carmel, "Time-to-completion in software package startups," in *1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, 1994.

- [5] E. Klotins, M. Unterkalmsteiner, and T. Gorschek, "Software engineering knowledge areas in startup companies: a mapping study," in *International Conference of Software Business*. Springer, 2015, pp. 245–257.
- [6] M. Unterkalmsteiner, P. Abrahamsson, X. Wang, A. Nguyen-Duc, S. Shah, S. S. Bajwa, G. H. Baltes, K. Conboy, E. Cullina, D. Dennehy *et al.*, "Software startups—a research agenda," *e-Informatica Software Engineering Journal*, vol. 10, no. 1, 2016.
- [7] T. Dyba, B. A. Kitchenham, and M. Jorgensen, "Evidence-based software engineering for practitioners," *IEEE software*, vol. 22, no. 1, pp. 58–65, 2005.
- [8] B. A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in *Proceedings of the 26th international conference on software engineering*. IEEE Computer Society, 2004, pp. 273–281.
- [9] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham, "Using mapping studies in software engineering," in *Proceedings of PPIG*, vol. 8. Lancaster University, 2008, pp. 195–204.
- [10] B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn, 2007.
- [11] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.
- [12] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.
- [13] T. Dyba, T. Dingsoyr, and G. K. Hanssen, "Applying systematic reviews to diverse study types: An experience report," in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. IEEE, 2007, pp. 225–234.
- [14] M. Ivarsson and T. Gorschek, "A method for evaluating rigor and industrial relevance of technology evaluations," *Empirical Software Engineering*, vol. 16, no. 3, pp. 365–395, 2011.
- [15] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. ACM, 2014, p. 38.
- [16] P. Tingling and A. Saeed, "Extreme programming in action: a longitudinal case study," *Human-Computer Interaction. Interaction Design and Usability*, pp. 242–251, 2007.
- [17] D. Martin, J. Rooksby, M. Rouncefield, and I. Sommerville, "'good' organisational reasons for 'bad' software testing: An ethnographic study of testing in a small software company," in *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society, 2007, pp. 602–611.
- [18] L. Hokkanen and K. Väänänen-Vainio-Mattila, "Ux work in startups: current practices and future needs," in *International Conference on Agile Software Development*. Springer, 2015, pp. 81–92.
- [19] A. N. Duc and P. Abrahamsson, "Minimum viable product or multiple facet product? the role of mvp in software startups," in *International Conference on Agile Software Development*. Springer, 2016, pp. 118–130.
- [20] T. Langer and P. Vaněček, "Agile methods in tech-startup," *IMEA 2012*, p. 63, 2012.
- [21] E. Lindgren and J. Münch, "Raising the odds of success: the current state of experimentation in product development," *Information and Software Technology*, vol. 77, pp. 80–91, 2016.
- [22] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: the greenfield startup model," *IEEE Transactions on Software Engineering*, vol. 42, no. 6, pp. 585–604, 2016.
- [23] C. Giardino, M. Unterkalmsteiner, N. Paternoster, T. Gorschek, and P. Abrahamsson, "What do we know about software development in startups?" *IEEE software*, vol. 31, no. 5, pp. 28–32, 2014.
- [24] G. Coleman and R. V. O'Connor, "An investigation into software development process formation in software start-ups," *Journal of Enterprise Information Management*, vol. 21, no. 6, pp. 633–648, 2008.
- [25] A. Miski, "Development of a mobile application using the lean startup methodology," *International Journal of Scientific & Engineering Research*, vol. 5, no. 1, pp. 1743–1748, 2014.
- [26] M. Taipale, "Huitale—a story of a finnish lean startup," in *Lean Enterprise Software and Systems*. Springer, 2010, pp. 111–114.
- [27] J. Björk, J. Ljungblad, and J. Bosch, "Lean product development in early stage startups." in *IW-LCSP@ ICSOB*, 2013, pp. 19–32.
- [28] J. Škrabálek and C. Böhm, "Why modern mobile and web-based development need a lean agile web approach (law)," *IDIMT-2013*, p. 225, 2013.
- [29] H. Terho, S. Suonsyrjä, A. Jaaksi, T. Mikkonen, R. Kazman, and H.-M. Chen, "Lean startup meets software product lines: Survival of the fittest or letting products bloom?" in *SPLST*, 2015, pp. 134–148.
- [30] S. Jansen, S. Brinkkemper, I. Hunink, and C. Demir, "Pragmatic and opportunistic reuse in innovative startup companies," *IEEE software*, vol. 25, no. 6, 2008.
- [31] R. Nanavati, "Experience report: a pure shirt fits," in *ACM Sigplan Notices*, vol. 43, no. 9. ACM, 2008, pp. 347–352.
- [32] E. S. De Moura, M. R. Herrera, L. Santos, and T. Conte, "When software impacts the economy and environment," *IEEE Software*, vol. 33, no. 6, pp. 23–26, 2016.
- [33] M. Taheri and S. M. Sadjadi, "A feature-based tool-selection classification for agile software development." in *SEKE*, 2015, pp. 700–704.