Proposta para Grupo de Trabalho 2023

Programa de P&D Serviços Avançados

Fase 2

GT-SecDeVIaS - DevSecOps as a Service

Cesar Augusto Cavalheiro
Marcondes Proponente
cmarcondes@ita.br

São José dos Campos - SP, 23 de setembro de 2022

1 Título

GT-SecDeVIaS - DevSecOps as a Service

2 Coordenador Acadêmico

Cesar Augusto Cavalheiro Marcondes
Professor do Instituto Tecnológico de Aeronáutica (ITA)
Lattes: http://lattes.cnpq.br/4431183539132719
LinkedIn: https://www.linkedin.com/in/cesarmarcondes/
Google Academico: https://scholar.google.com.br/citations?hl=pt-BR&user=VqUfaJsAAAAJ
Dados de Contato: +55 (16) 997193230 (celular com whatsapp), email: cesar.marcondes@gp.ita.br

3 Assistente de Inovação

Eng. Manuel Correia
Fundador da Netconn Group, executivo de várias empresas TI nos últimos anos e participante da Rede ANSP.br onde foi o responsável pela montagem do seu ecossitema de inovação.
Site da Startup: https://www.netconngroup.com.br/
LinkedIn: https://www.linkedin.com/in/mcorreia/
Dados de Contato: +351 913 869 261 (celular com whatsapp), email: mcorreia@gmail.com

4 Tópicos de Interesse

DevOps, DevSecOps, Segurança Cibernética, Desenvolvimento de Software Seguro, Pipelines de Segurança,
Análise de Segurança de Código, Computação em Nuvem

5 Parcerias e respectivas contrapartidas

- O Instituto Tecnológico de Aeronáutica (ITA) é uma instituição de ensino superior pública da Força Aérea Brasileira, vinculada ao Departamento de Ciência e Tecnologia Aeroespacial (DCTA), localizado na cidade de São José dos Campos, São Paulo. É considerado uma das melhores instituições de ensino superior do Brasil, com formação em engenharias, incluindo a engenharia da computação. A instituição tem se destacada nos últimos anos na área de segurança cibernética com acordos com empresas e com o Comando de Defesa Cibernético (CComDCiber). O projeto será conduzido no ITA, dando preferencia para a contratação de alunos da instituição.
- O Exercito Brasileiro tem a atividade estratégica de ciberdefesa, através de seus vários órgãos como CDS, ComDCiber e ENADCiber tem prerrogativa de proteger os sistemas de informações e neutralizar a fonte de ataques, tentando inibir possíveis ataques digitais. Integra o Sistema Militar de Defesa Cibernética, que atua em cinco áreas de competência: Doutrina, Operações, Inteligência, Ciência e Tecnologia e Capacitação de Recursos Humanos.

- A NetConn Group será responsável pela divulgação e comercialização. Ela irá dedicar horas de trabalho nessas atividades e também o provisionamento de recursos computacionais de produção para o uso pelos early adopters, a custo zero para o projeto. Ela tem atuação no Brasil e em Portugal através do seu diretor-geral Manuel Correia.

6 Descrição da Proposta

6.1 Sumário Executivo

O mercado de DevOps, segundo relatorio da Gitlab de 2022[1] vem amadurecendo rapidamente, passando de 16% em 2021 para 47% dos times internacionais fazendo algum tipo de automação DevSecOps, e os desenvolvedores tem recebido cada vez maior responsabilidade pela implantação automática de sistemas, com a segurança andando de mãos dadas com o ciclo de desenvolvimento. A velocidade de entrega, também aumentou muito, cerca de 70% dos projetos são revisados e implantados com *commits* diariamente. Outro ponto interessante é que em relação a 2021, o número de times que usa alguma forma de aprendizado de máquina para revisão do código pulou de 15% para 31% devido a aceleração de entregas, muito mais rápidas. Nesse contexto, o projeto GT-Devias Fase 1, que estamos renomeando para SecDeviaS Fase 2 esta muito alinhado com o mercado.

Do ponto de vista técnico, o MVP Fase 1 do SecDevias esta praticamente concluído fornecendo um ambiente de DevSecOps automatizado para clientes, recebendo submissões de códigos e repositórios e criando uma auto-esteira com base em várias ferramentas estado da arte de segurança de código, e produzindo posteriormente relatórios e timestamp auditável, que ajudam os desenvolvedores a entender e corrigir ataques na camada de software. O MVP foi desenvolvido para ser escalável, e tem instancias rodando em nuvem OpenStack em datacenter e AWS, além disso, sua integração com as plataformas da RNP esta sendo desenvolvida com a autenticação federada com CAFe, e a entrada no funil de serviços no NasNuvens da RNP é planejado para a Fase 2.

Em termos de amadurecimento de negócio, fizemos muitas reuniões com empreendedores, diretores de times de desenvolvimento e academia, para tentar encontrar a aderencia correta para a comercialização. Os resultados dos dialogos indicam interesse, mas precisamos entregar algo a mais que as ferramentas de maneira isolada já não ofereçam. Por isso, a proposta para a Fase 2 foca na evolução tecnológica da nossa auto-esteira para incluir um novo algoritmo de *deep learning* completamente diferente do mercado, com interpretabilidade (algo inédito, mesmo em termos de estado da arte) e também em novos oferecimentos na esteira como DAST (incluindo API), detecção de licenciamento, patches de dependencias e finalmente, refatoração automática por meio de receitas de consertos. Esses desenvolvimentos irão propiciar a diferenciação do produto. Em paralelo, evoluiremos o MVP da Fase 1, construíndo-o com maior escalabilidade, e refatoração do *frontend* para ir atendendo os pedidos que forem realizados de clientes.

A diferenciação permitirá a exploração de diferentes modelos de negócio, com valores agregados mais alinhados não somente com a detecção de vulnerabilidades, mas também as sugestões apropriadas de correções. Por exemplo, poderá ser oferecido valores diferenciados para: o *freemium*, varredura mais

[1] https://about.gitlab.com/developer-survey/

profunda baseado em regras, varredura baseada em IA, e impulsionar medição e precificação de correções aumentando a escalabilidade com auto-correções. Desse modo, como os times de desenvolvimento tendo, cada vez, menos tempo para corrigir problemas, isso dará flexibilidade para a tomada de decisão sobre como atuar na detecção, como filtrar o que é realmente importante, e o que corrigir. Por último, o projeto fazendo parte do funil de oferecimento do NasNuvens, poderá ser ofertado, tanto para os times especializados em desenvolvimento de software da RNP como o CAIS Pentesting, e o USDE de projetos digitais, quanto também para as universidades, e para o mercado de Pequenas e Médias Empresas ajudando a aumentar a maturidade de segurança do setor de software brasileiro.

6.2 Desenvolvimento Tecnológico

O MVP desenvolvido na Fase 1 da solução SecDeVIaS (Figura 1) visava produzir uma solução que fornecesse, dentro de uma perspectiva automatizada da cultura de DevSecOps, relatórios de segurança indicando as vulnerabilidades encontradas, e recebendo feedback do usuário sobre o processo. O MVP ajuda a dois tipos de usuários, desenvolvedor e gerente de produto (ou lider de desenvolvimento) a ter uma visão holistica dos problemas encontrados de segurança em código, e consequentemente, diminuir a superfície de ataque ao software.

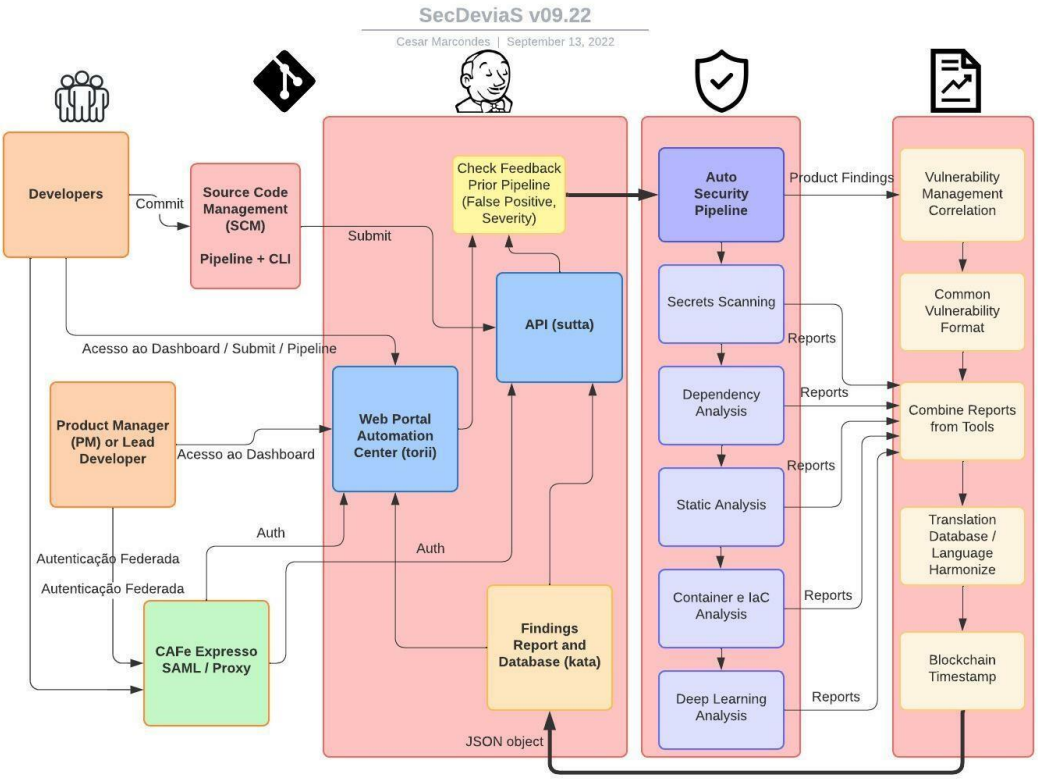

Figura 1: MVP desenvolvido pela Fase 1 do GT.

De maneira geral, ao longo do projeto Fase 1, a arquitetura do MVP foi mudando em função de atualização dos requisitos. Na Figura 1, iniciamos com os clientes em laranja. Para ter acesso a plataforma, pode-se cadastrar o usuário por email, ou, em verde, através da integração com autenticação federada da RNP (em desenvolvimento). Em seguida, em azul, temos dois modos de acesso à solução - via API ou portal web, destaco que a API facilita ao desenvolvedor não perder o foco, e pode ser acionada diretamente sem acesso web. No MVP tem-se a possibilidade de feedback embarcado em botões, permitindo que clientes

marquem falsos positivos e alterem padrão de severidade, ou feedback de textos. Isso é adicionado à auto-esteira de segurança, carro-chefe do SecDevias. Essa é feita de ferramentas de segurança de código e é acionada periodicamente a cada commit/submit. As ferramentas de segurança utilizadas são todas open-source, com licenças permissivas para comercialização. Destaca-se que foi incorporado ao processo de DevSecOps automatizado, os seguintes tipos de varreduras: verificação de credenciais no código (Secrets), verificação de bibliotecas com vulnerabilidades (Dependency), aplicação de regras em código estático baseado na maioria das linguagens de programação (SAST) e também adicionado avaliação de artefatos baseados em containers e infraestrutura como código, (ex. scan de códigos de terraform). A parte final do MVP é outro backend que realiza a gestão e correlação das vulnerabilidades, passando pela uniformização do formato das descobertas, e posterior auto-tradução via nuvem. Finalmente, todo o processo é sacramentado com um timetamp eterno, a custo zero na Blockchain BTC [4]. E o retorno é enviado como um objeto JSON, formatado apropriadamente no portal web e serviço de API.

Tecnologicamente, o atual MVP tem clara diferenciação na promoção de uma auto-esteira de DevSecOps sem a necessidade de intervenção humana. Em geral, ao analisar o oferecimento de concorrentes internacionais como AWS Pipelines, Azure DevOps Services, Sonarqube Cloud e Gitlab e Github Security, nota-se o foco somente em, prover uma infraestrutura em nuvem para execução de esteiras e analises. Porém, pouca preocupação com a confecção da esteira em si, deixando isso com o time, ou contratação de um DevSecOps humano, por exemplo. No Brasil, as empresas nacionais pesquisadas: InsiderSec.io, BugScout.io, ScanX.ia e Yaman, oferecem serviços de consultoria manual especializada, para a criação de esteiras, um negócio caro e pouco escalável.

De modo geral, o Gitlab Ultimate é o que mais se aproxima do nosso MVP, porém sem a parte de tradução e sem o gerenciamento do ciclo de vida de vulnerabilidades, o feedback e a correlação. Tudo isso, ao custo de 100 dolares/mes por repositório. Na Fase 2 do projeto pretendemos investir mais na diferenciação dos tipos de varredura de vulnerabilidade. A intenção é usar algoritmo de aprendizado de maquina (CoTexT) baseado em NLP, combinado com interpretabilidade, ao invés do uso somente de ferramentas baseadas em regras, como o Github Ultimate faz. Além disso, pretendemos oferecer outras formas de varredura maciças utilizando-se de DAST e algo similar a *queries* de CodeQL. Finalmente, pretendemos também nos diferenciar no provimento de “serviços” escaláveis de consertos de vulnerabilidades através da acoplamento na auto-esteira com software livre de refatoração em larga escala (OpenRewrite). Atualizando tanto bibliotecas, quanto patches em bugs de CVEs recentes, conforme Arquitetura descrita na Figura 2. Na figura, o software de MVP Fase 1 esta compactado (torii, kata e sutta) no centro (quadrado vermelho), e a auto-esteira com gerencia de correlação, bem como os novos elementos planejados na Fase 2, serão incorporados à auto-esteira (em azul escuro).

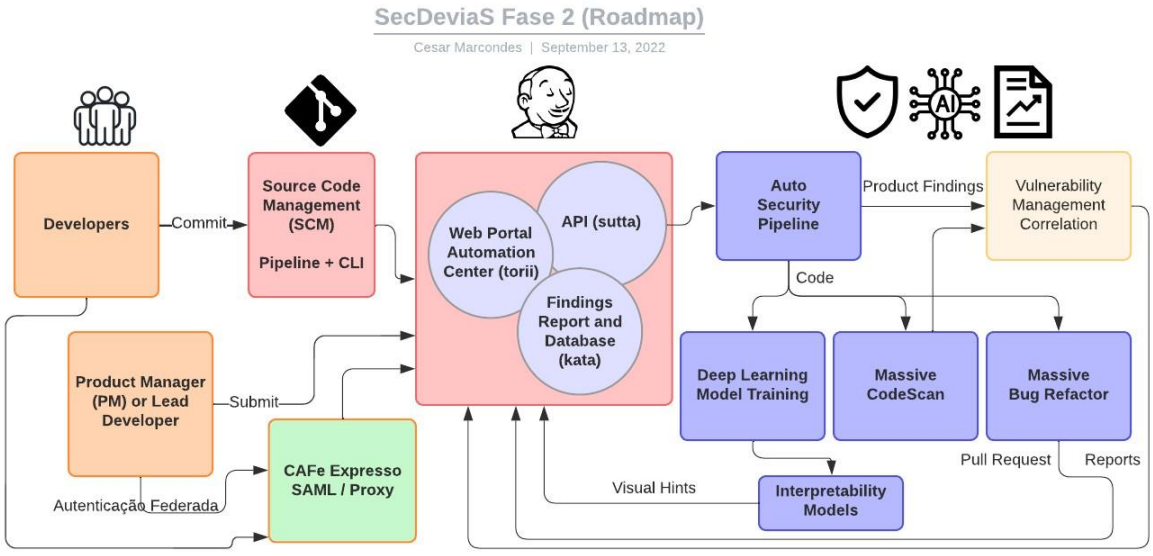

Figura 2: Evolução do MVP na Fase 2 do GT.

Desse modo, em termos da evolução na Fase 2 do produto, embora seja desejável a melhoria em termos de escalabilidade e de oferta em nuvem, entretanto, uma parte considerável disso já estará concluída na Fase 1.

Por exemplo, reportamos à RNP testes de carga realizados de 100-500 pipelines concorrentemente, e a nossa tecnologia baseada em kubernetes permite a escalabilidade horizontal e vertical atendendo número de scan de repositórios escalável. Assim sendo, embora a nossa arquitetura cloud-native continue evoluindo, mas o foco na Fase 2 é na diferenciação do produto com maior volume de treinamento de modelos de aprendizado de maquina para o algoritmo NLP, realizando uma constante fase de aprendizado e submissão, aumentando o escopo das buscas de vulnerabilidades, e incorporando refatorações automáticas na própria esteira, conforme descrito na arquitetura.

A tecnologia de Deep Learning Model Training sendo implementada no MVP do SecDevias é baseada no CoTexT [3]. Para tanto, implementamos o nosso próprio código do modelo, em python, e fizemos uso dos pesos do *transformer (Code to Text)* do modelo original (treinado em 6 linguagens de programação: python, java, javascript, php, ruby e go). Em seguida, conforme descrito na literatura, fizemos o ajuste fino com uma cópia do *dataset* base de benchmark CodeXGLUE, reproduzindo comparações com o estado da arte. Esse *dataset* de referencia possui grande número de funções vulneráveis em C/C++. E os nossos resultados foram muito próximos do melhores reportados no benchmark CodeXGLUE no quesito Defect Detection (Code-Code) [2] com F-score de 0.56 e acurácia de 0.65. Em outras palavras, indicando resultados superiores aos métodos baseados em regras, na redução de falsos positivos. Além desse modelo, também foram testados outros tantos modelos (Fase 1), a maioria com base em codificação de redes neurais em grafos, mas apresentando resultados inferiores, como o GNN-ReGVD e o GNN-Devign. Na Fase 2 do GT, o modelo CoTexT poderá ser re-treinado multiplas vezes, e por meio de ajuste fino específico para cada linguagem, utilizando-se por exemplo, de repositórios públicos do github para java, dotnet e python e fazendo o scan com nossas próprias ferramentas de detecção de vulnerabilidades, por meio de regras do MVP Fase 1, para

[2] https://microsoft.github.io/CodeXGLUE/

popular novos *datasets* de treinamento específico, pois como o CodeXGLUE é para C/C++. Assim, teremos o equivalente para python, java, etc. Esses testes exaustivos, serão feitos na testbed da startup NetConn Group e posteriormente, será utilizada VM com GPU e ferramenta de deep learning para produzir modelos cada vez mais especializado por linguagem da Fase 2.

Outro ponto importante, na tecnologia de IA é que métodos baseados em redes neurais, embora eficientes e fazem correlações que os humanos tendem a não enxergar, tem o inconveniente de serem métodos de caixa-preta. Portanto, só trazem detecção no nível individualizado das funções. Em outras palavras, dizem se cada função do código é vulnerável ou não, mas não trazem qual linha específica é vulnerável, diferente dos métodos de regras. Desse modo, para aumentar a produtividade do desenvolvedor em encontrar a linha de código vulnerável, estará sendo aperfeiçoado na Fase 2, uma forma de extrair Interpretability Models a partir dos pesos da inferencia. Um levantamento mostrou que o método de interpretabilidade com melhor desempenho para aplicações de processamento de linguagem natural utilizando-se de transformers (que é o caso do CoTexT) é o InputX Gradient com Distancia Euclidiana de L2 para agregação [1]. Resultados preliminares foram desenvolvidos (Fase 1) usando InputX Gradient e nos permite dar uma noção de onde o modelo deriva sua decisão por códigos vulneráveis. Na Figura 3, a seguir, realizamos uma conversão dos parametros obtidos pelo InputX para um heat-map, ilustrando partes do código interpretadas como mais importantes para o modelo de detecção de defeitos (CoTexT) com nossos parametros. É possível notar, por exemplo, que o nosso modelo tem preferencias por ponteiros amorfos em C (como o void * opaque), e também em vetores com possíveis limites vulneráveis (como o cpu_ir[ra]), bem como parametros de decisão nos if-statements (if (islit)). Essa visualização estará nos relatórios do SecDevias para o usuário ficando disponível por meio do portal web, e API, para ajudar a investigação pelos desenvolvedores.

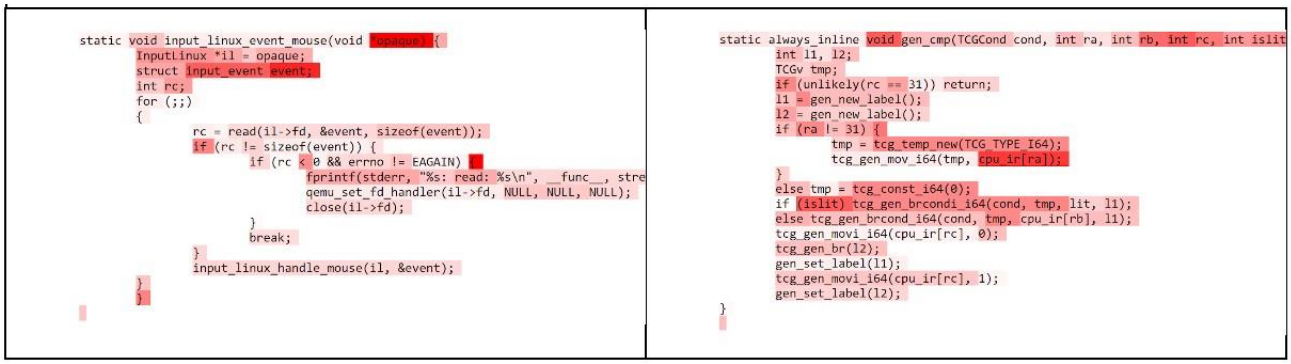

Figura 3: HeatMap Preliminar de Interpretabilidade do Modelo CoTexT na esteira do SecDevias

Outros caminhos da evolução (roadmap) da Fase 2 do SecDevias (Figura 2 - Massive CodeScan e Massive Bug Refactor) incluem novos modos de varreduras na esteira, e possibilidade de refatoração automática. A esteira de DevSecOps é um processo cultural nas empresas e ela pode ficar desatualizada rapidamente, como um projeto ad-hoc sem manutenção, desse modo, o uso da auto-esteira pelo SecDevias empodera os clientes e traz o estado da arte para eles, com o mínimo de esforço.

Portanto, no Massive CodeScan, além de já ter incorporado SECRETS, SAST, SCA (security component analysis) e CONTAINERS e IaC, para a Fase 2, vamos incorporar outros modos: a) DAST (que é o scan dinâmico) onde o cliente informará um *endpoint* onde foi feito o *deploy* do código (incluindo API) e executará fuzzing para encontrar outras brechas, com relatórios traduzidos, mesmo com *web firewall*. b)

License compliance baseado em open-source, faz comparação do código com github e detecta uso de licenças restritivas. c) *Dependabot* também baseado em open-source faz o processo automático, na esteira, de sugestão de atualização de dependencias. E finalmente, d) *Queries* em CodeQL [2]. Faremos testes academicos com o CodeQL, ou tentaremos implementação própria de CodeQL (ou licenciamento). No CodeQL, os códigos são escaneados utilizando-se da linguagem de busca (uma espécie de SQL para código) podendo descobrir outros tipos de vulnerabilidades. Existem muitas regras disponíveis, e podemos fazer mais nessa frente.

O ultimo item de evolução é a combinação de humano/máquina para resolver bugs e refatorar código (Massive Bug Refactor). Para tanto, no pipeline do SecDevias vamos adicionar um botão para que o cliente faça um pedido de ordem de serviço para *bug fix*. Esse pedido vai acionar uma Service Helpdesk interno que vai escalonar algum membro (humano) da startup para corrigir o código no limite de remover o alarme. Obviamente, não é possível dar garantias que um código limpo não tenha vulnerabilidades não-detectadas pelas nossas ferramentas, então, vamos procurar diminuir nossa responsabilidade. Como o conserto de vulnerabilidade de um cliente pode ter repercussão semelhante na base de outros, a ideia nesse espaço é criar modelos próprios de refatoração maciça usando-se da ferramenta OpenRewrite [3], desenvolvida originalmente na Netflix e colocada em open source. O OpenRewrite permite escrever de maneira programatica e navegar em AST (Abstract Syntax Tree) de códigos, modificando de maneira genérica por meio de templates, as chamadas receitas, onde trechos vulneraveis de código trocam por outros trechos corretos. Portanto, as intervenções humanas podem gerar novas "receitas" OpenRewrite *in-house* que podem ser aplicadas em todos os repositórios de clientes através da auto-esteira.

6.3 Evolução do Modelo de Negócios

Durante a submissão do projeto da Fase 1 do MVP, fizemos uma proposta inicial de oferecimento do negócio como SaaS (Software como Serviço) na modalidade de *Freemium* (tendo acesso grátis à algumas varreduras, e pago para o completo) porém após quantidade grande de reuniões com potenciais clientes e verificando os interesses em pagamentos, temos a possibilidade de tentar outros modelos de negócio em vista na Fase 2. Como retrospectiva da Fase 1, fizemos nos primeiros meses do projeto cerca de 47 entrevistas com potenciais clientes, divididas entre ecossistema da RNP, setor público e setor privado. Entre as lições aprendidas, muitos consideraram que poderiam usar o produto que era inovador, mas não saberiam quanto custaria, e isso gerou uma indicação da necessidade de se estabelecer *early adopters*. Nesse sentido, foram incorporados *feedbacks* dos potenciais clientes e RNP ao longo do projeto, para ficar mais compatível com as necessidades que são mais importantes. No caso da RNP, nos reunimos com vários setores internos da RNP (USDE, P&D, CAIS) para entender o seu processo de desenvolvimento, e se o MVP ajudaria a tornar mais maduro e seguro esse processo para a RNP, pensando nela como um cliente.
Nas reuniões sentimos a necessidade de criar uma API, sem intervenção com o portal Web.

Do lado comercial, fizemos reuniões com vários contatos da StartUp. Em especial, tivemos conversas com a empresa Altio[4], cujo sócio-fundador foi CIO da Porto Seguro (uma das maiores empresas de seguros do Brasil) com ampla experiencia comercial na área de TI. A Altio tem interesse em fazer testes com a ferramenta, mas também nos abriu as portas através de convites, para participar de algumas reuniões na

[3] https://github.com/openrewrite/rewrite
[4] https://altio.com.br/pt-br/

INOVABRA. A INOVABRA é a incubadora ligada ao Banco BRADESCO, e fizemos reunião com a líder de ecossistema da instituição. Entre os feedbacks que obtivemos, vimos que o INOVABRA não tem muitas empresas de segurança, as poucas listadas trabalham com *pentest*. A líder do ecossistema ficou interessada em como a tendencia atual de *no-code* poderia interferir no futuro de varredura de código e na auditoria por *timestamp* na blockchain. Também fomos convidados a participar de duas edições do evento interno da INOVABRA chamado "Pizza com CEOs", onde interagimos com diretores de várias startups e mesmo de grandes instituições. O feedback, em geral, foi que as startups não tem esteiras, porém não se preocupam com seu código, jogando toda a responsabilidade em WAFs (web application firewalls) em nuvem. E as grandes empresas tem esteiras, mas é um processo muito fragmentado, desatualizado e sem uniformização. Esses *feedbacks* ajudaram a evoluir o planejamento de *scan* para adicionar DAST (com WAFs *bypass*), conforme tecnologia da Fase 2. Outros contatos importantes do lado comercial foram com a ACADI-TI[5], responsável por cursos de cibersegurança com certificação. Para a ACADI-TI existe o interesse em conversar com parceiros de fábricas de software, ou mesmo, incorporar o SecDevias como uma ferramenta a ser usada em cursos de certificação de DevSecOps ou mesmo AppSec. Entretanto, as conversas não evoluiram por falta de MVP na época. Outro contato foi com o Gerente de Segurança da Farfetch (empresa de Moda Luso-Britanica, de Portugal) onde nós recebemos *feedback* mais técnico sobre a qualidade da tradução, sobre a possibilidade de trocar a severidade. E finalmente, foram feitos contatos com a ABES (Associação Brasileira das Empresas de Software) para a realização de um seminário, porém o custo era muito elevado (cerca de 7 mil reais) naquele momento e arriscado sem MVP.

Todos esses contatos de 2022 (RNP USDE, RNP CAIS, UNESP, ALTIO, INOVABRA, ACADI-TI, Farfeth, ABES) serão continuados em 2023. Portanto, a estratégia é fazer PoCs com esses potenciais clientes com a finalidade de testar a varredura por regras via API e web. Nesses casos, em termos de modelos de negócios que estão sendo vislumbrados, o PoC poderia ser oferecido como uma mera avaliação e *feedback* dos clientes, nesse grupo bem pequeno e restrito de "amigos" testadores. Outra estratégia mais agressiva de mercado, poderia ser o *Try-and-Buy* onde é fechado um acordo comercial leve, onde o cliente paga minimamente pelo PoC, e ao mesmo tempo, se compromete que se forem atendidos os seus pedidos, que ele assina por tempo mais longo (exemplo, anual). Esses são modelos de negócio mais tradicionais, onde a startup teria que colocar um esforço extra, para cada contrato, para atender adequadamente clientes, e não é escalável como SaaS. Esses PoCs com "amigos" ou pagantes serão feitos ao longo dos 12 meses do projeto para a conclusão das validações do MVP e sua evolução. E poderá ser ofertada para outras empresas e mesmo universidades e dentro do ecossistema da RNP, conforme forem avançando os testes de validação.

Com relação ao modelo de negócios *Freemium*, ele pode trazer grande atração de clientes gerais, por divulgação na Internet. Nesse caso, a ideia seria dar a chance de 5 execuções da esteira por cliente / projeto por mes, com o retorno limitado de vulnerabilidades críticas e altas. E quando o modelo de *deep learning* estiver mais maduro, oferecer pela internet também o scan por *deep learning*. Finalmente, no oferecimento público pela Internet, adicionar a possibilidade de estabelecimento de pacotes de auxilio manual (combinando humano e maquina na configuração da auto-esteira), para filtragem de falsos positivos, e também conserto e refatoração de vulnerabilidades críticas, segundo um custo ainda a ser estudado. Assim como acontece com jogos de celular, o oferecimento grátis será combinado com *in-game purchases*, ou compras dentro do jogo. Portanto, a ideia de evolução do modelo *Freemium* é deixar os botões de *feedback*

[5] https://acaditi.com.br/

do MVP acionáveis com custos variados, ex. 2000 reais/bug fix. Mas isso pode ser alterado a medida que mais testes de pivotação de modelo de negócios são feitos e custos mais precisos sejam estimados.

Conforme, já haviamos apontado anteriormente na Fase 1, a união de pesquisadores de cibersegurança de instituições militares (ITA, e EB), e do retorno de informações de como lidar com grandes projetos como a RNP (USDE) e em conjunto com a startup Netconn, especializada na interação Universidade - Empresas, tem dado um bom resultado de visibilidade e avanço tecnológico e de negócios, conforme evidenciamos aqui. As parcerias continuarão como estavam, com o desenvolvimento tecnológico sendo produzido pelos bolsistas das instituições e a startup procurando oportunidades de novos negócios e o crescimento do mesmo. Também destacamos que embora a maiorias dos concorrentes tenham como alvo: grandes e consolidadas empresas, pelo alto retorno financeiro por projeto dessas empresas. Ainda assim, continuamos com o intuito de trazer um produto que possa ser adquirido por pequenas e médias empresas, e fábricas de software, pelo fatiamos dos oferecimentos em diversos pacotes. Do lado da RNP, já esta sendo desenvolvido a integração com a autenticação federada e poderá ser oferecido o pacote via NasNuvens da RNP, na modalidade *Freemium* com *in-game purchases* para todo o ecossistema onde a RNP presta serviço.
Seja universidades, governo federal ou centros de pesquisa, tudo a custo marginal baixo e facilidade de uso.

Conforme detectamos, com base na nossa tecnologia inovadora, será preciso um esforço para capturar os *early adopters* nas personas dos usuários desenvolvedor e lider de projeto. A persona do desenvolvedor não desembolsa esse tipo de ferramenta para seu uso pessoal, em geral, é uma decisão empresarial, mas ele que usa, divulga, apoia e adota, então é importante ter o desenvolvedor como o usuário que chega no site web e faz o teste do produto. A outra persona é o líder de projeto, com base no *dashboard* produzido no MVP Fase 1 é possível chamar sua atenção para uma visão holística da empresa, os tipos de vulnerabilidades sendo produzidos e, futuramente, mostrar quais os desenvolvedores que são os que mais introduzem vulnerabilidades. Note que e em geral, ele é ordenador de despesas. Finalmente, uma terceira persona é o time de segurança, que em geral, gerencia o orçamento de segurança de TI, é preciso convence-los a divulgar o produto como uma camada de proteção, estado da arte, importante para a empresa.

7 Cronograma de marcos

O projeto estará organizado utilizando-se de uma estratégia de desenvolvimento ágil, em 12 *sprints*. Cada *sprint* terá duração fixa de 1 mes. No desenvolvimento ágil, cada *sprint* é composto pelo desenvolvimento, teste e validação de um conjunto de estórias do produto. Serão feitas reuniões quinzenais com o time, no equivalente as *stand-up meetings* da metodologia Scrum. Os desenvolvimentos tecnológicos serão separados em 5 blocos de épicos: 1) Melhoria do MVP como um todo, 2) Máquina de Vendas e Atendimento aos Clientes, 3) Deep Learning, 4) Novas Modalidades de Scan e 5) Refatoração Automática. Cada bloco épico compreende a várias atividades de design, desenvolvimento e integração, e tenderá a ter um responsável bolsista em separado. Assim, as entregas vão sendo feitas ao longo dos 12 meses de projeto, e serão independentes uma das outras. Sub-divididas por linguagem, ou modalidades.

Epics	Descrição do Entregável

Melhoria do MVP (Fase 1)	Baseado nos usuários ao longo dos PoCs, como por exemplo, botão de feedback sobre qualidade do falso positivo, botão de feedback sobre a adequação de textos. Outras formas, de integração com a auto esteira, a partir do Jenkins, a partir de Circle CI. Métodos de retorno por meio de mensagens em Slack, Discord. Reformulação do layout do Dashboard. Refatoração do código. Criação de grupos de usuários representando instituições com projetos e repos organizados nesses grupos.
Máquina de Vendas e Atendimento aos Clientes	Desenvolvimento de escalonador de mensagens para redes sociais e campanhas de email, criação de ambiente de helpdesk online com tickets com prioridades, criação de funil de marketing com divulgação, e acionamentos específicos para clientes na jornada de acesso ao produto. Preparação de material de divulgação com *input* do time.
Evolução do Deep Learning	Evolução dos algoritmos e modelos de Deep Learning para detecção de falhas no código. Construção de *datasets* específicos para o ajuste fino por linguagem de programação, a partir da aplicação de regras sobre repositórios públicos do github. Evolução dos modelos de interpretabilidade e heatmaps, adicionar uma retroalimentação sobre os erros e acertos, com intervenção humana, para que os resultados dos modelos de interpretabilidade possam ajudar em melhor desempenho no modelo CoTexT principal.
Novas modalidades de *Scan*	Melhorar a auto-esteira com novas modalidades de scan, adicionando a possibilidade de fazer DAST (mesmo contra um *web firewall*, inclusive API). Adicionar *scan* de licenciamento, adição de atualização automática por meio de bot de dependencias, e finalmente testes com o CodeQL (Github) ou alternativo, para detectar vulnerabilidades por meio de *queries* especificas por CVE, algo apelidado de (Githubification of InfoSec).
Refatoração Automática	Adicionar a auto-esteira a possibilidade de refatorar automaticamente códigos por meio de receitas de OpenRewrite. Estudar sobre como seria o design da busca pela refatoração, montagem de exemplos de refatoração prontos e aplicar sob demanda nos repos dos clientes. Portar a refatoração do OpenRewrite (que é especifico para Java) para outras linguagens como python ou dotnet.

É importante ressaltar que poderemos estudar várias antecipacões nas entregas e marcos acima descritos, na medida da necessidade e interesse de clientes, o qual pretendemos analisar em conjunto com a equipe da RNP. Também cumpriremos o calendario da Fase 2, conforme acordado na carta convite.

8 Referências

[1] ATANASOVA, P., SIMONSEN, J. G., LIOMA, C., AND AUGENSTEIN, I. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 3256–3274.

[2] MOOR, O. D., VERBAERE, M., HAJIYEV, E., AVGUSTINOV, P., EKMAN, T., ONGKINGCO, N., SERENI, D., AND TIBBLE, J. Keynote address: .ql for source code analysis. In *Seventh IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM 2007)* (2007), pp. 3–16.

[3] PHAN, L., TRAN, H., LE, D., NGUYEN, H., ANNIBAL, J., PELTEKIAN, A., AND YE, Y. Cotext: Multitask learning with code-text transformer. *Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021)*.

[4] TODD, P. Opentimestamps: Scalable, trust-minimized, distributed timestamping with bitcoin, Sep 2016.