snyk

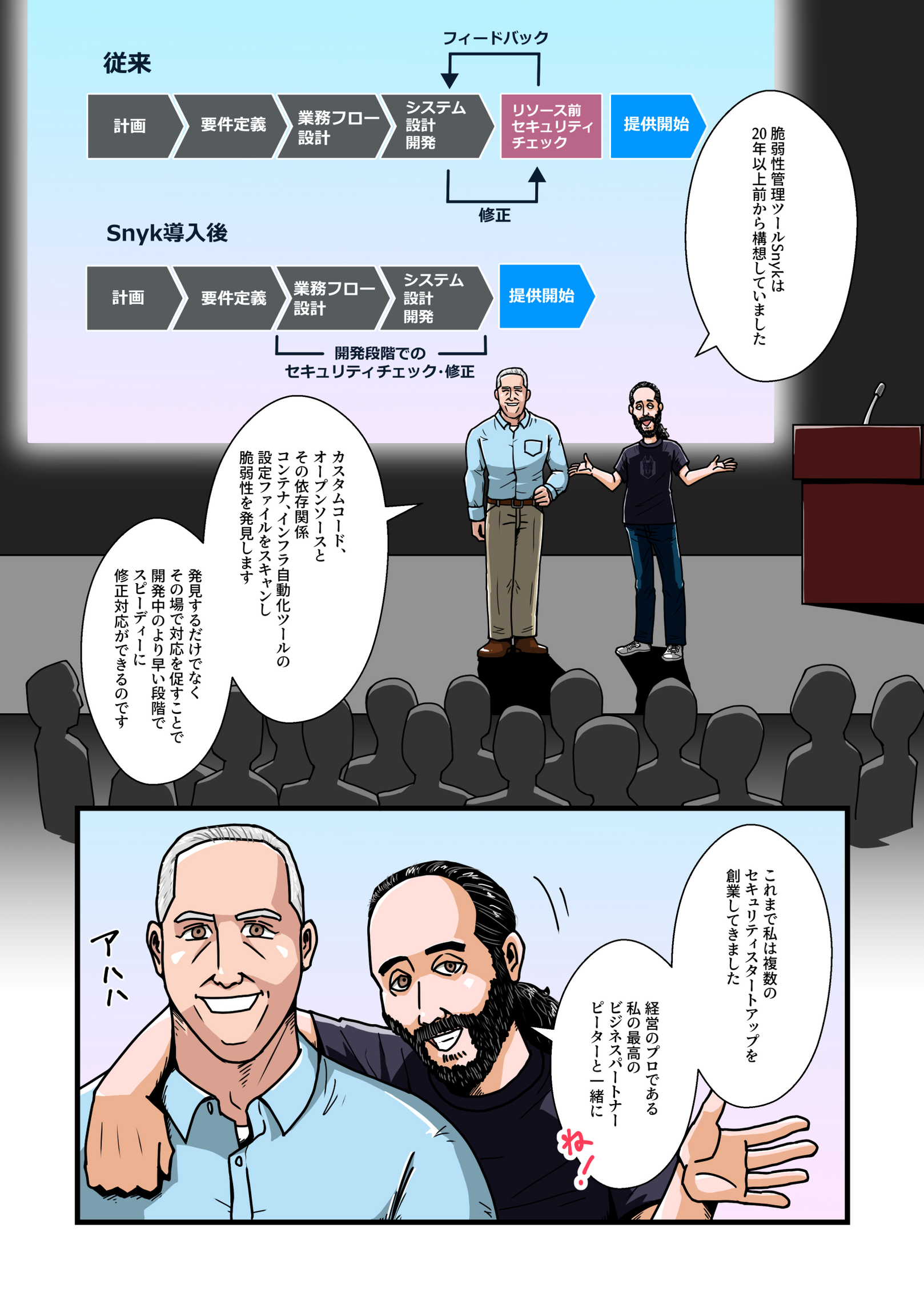


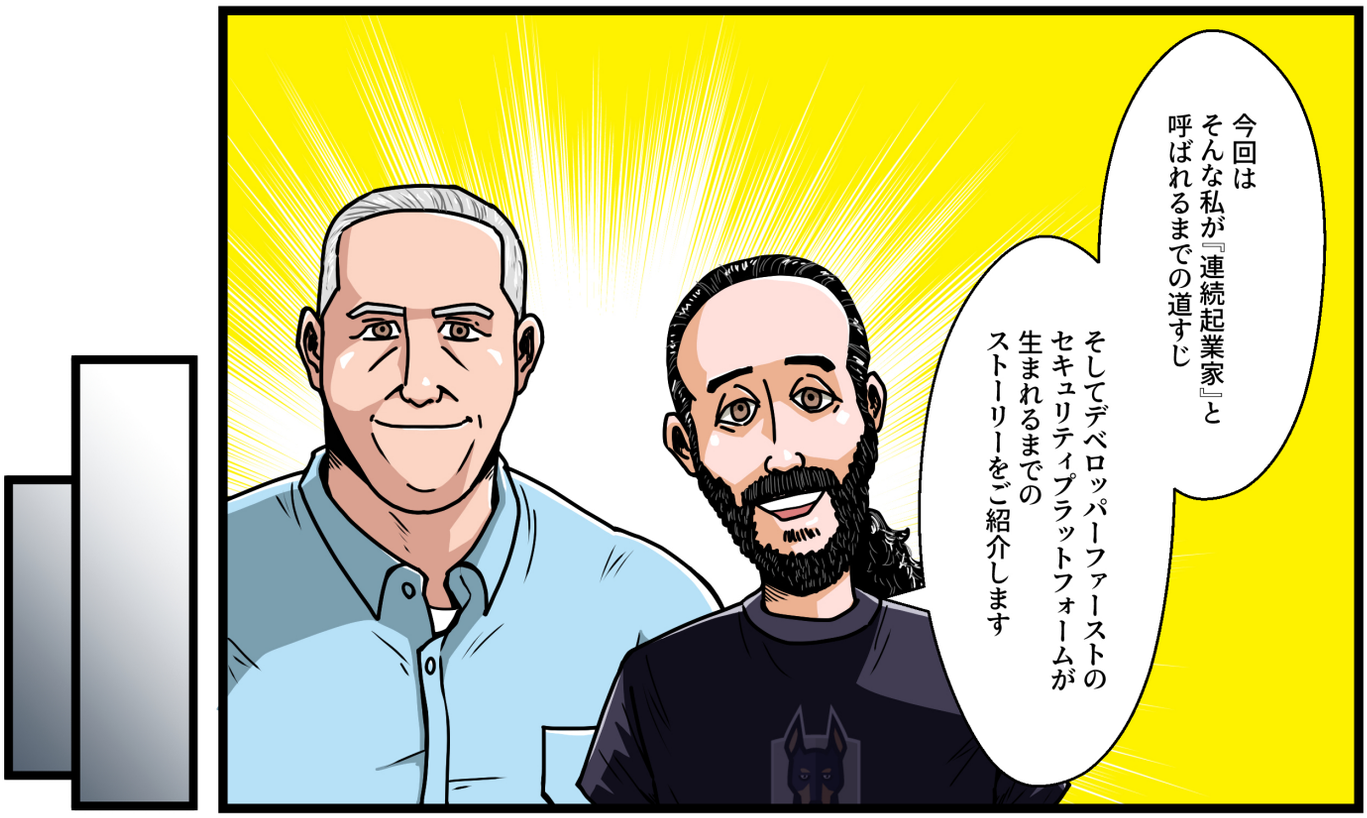
マンガでわかる!

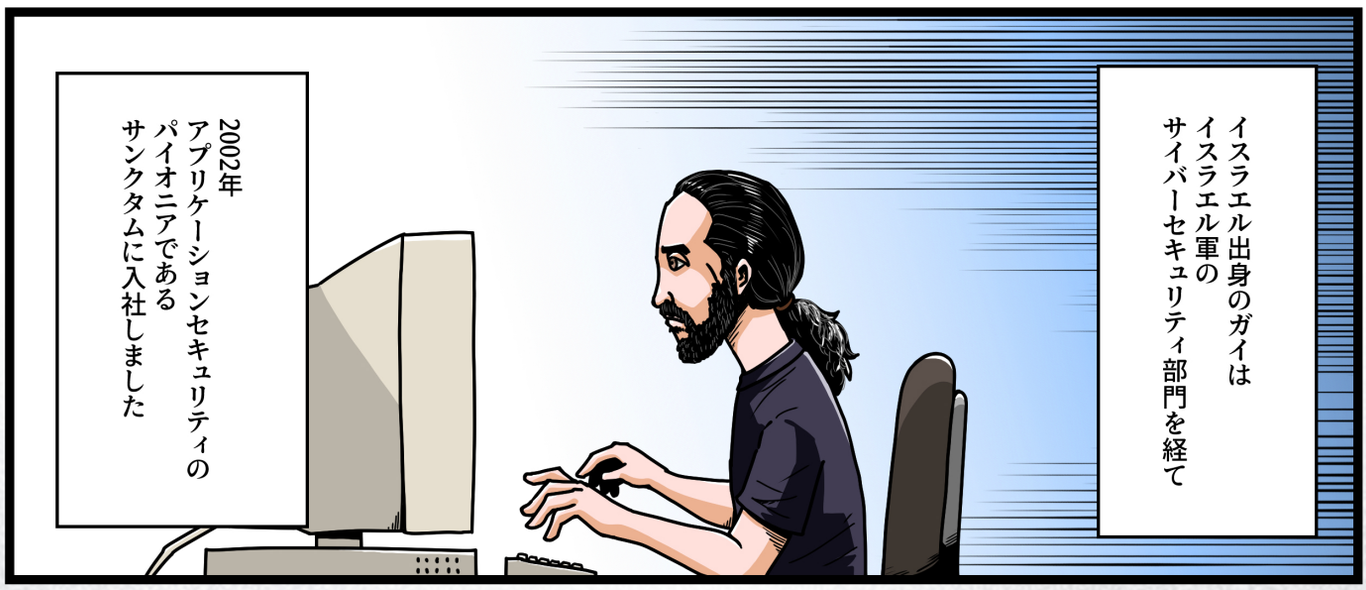
Snyk 誕生物語

Peter McKay and Guy Podjarny
ピーター・マッケイ　ガイ・ポジャーニー


Snykの創設者
ガイ・ポジャーニー
snyk
『連続起業家』とも
言われる彼は――
Snykは
デベロッパーファーストの
セキュリティプラットフォーム…
つまり
サイバーセキュリティの
企業でありながら
開発者に優しいという点を
核心としています
現在の開発において
必須条件となってきた
シフトレフト※において
開発者がセキュリティを
受け入れるしかない…
そんな状況を
改善すべく
私は立ち上がりました
snyk
※シフトレフト…短期化した開発ライフサイクルのなかで
リリースのスケジュールを妨げないよう、
セキュリティの行程を前倒しして実施すること


従来
フィードバック
計画
要件定義
業務フロー設計
システム設計開発
リソース前セキュリティチェック
提供開始
修正
Snyk導入後
計画
要件定義
業務フロー設計
システム設計開発
提供開始
開発段階でのセキュリティチェック・修正
脆弱性管理ツールSnykは20年以上前から構想していました
カスタムコード、オープンソースとその依存関係コンテナ、インフラ自動化ツールの設定ファイルをスキャンし脆弱性を発見します
発見するだけでなくその場で対応を促すことで開発中のより早い段階でスピーディーに修正対応ができるのです
これまで私は複数のセキュリティスタートアップを創業してきました
経営のプロである私の最高のビジネスパートナーピーターと一緒に
ね！
アハハ


今回は
そんな私が『連続起業家』と
呼ばれるまでの道すじ
そしてデベロッパーファーストの
セキュリティプラットフォームが
生まれるまでの
ストーリーをご紹介します


イスラエル出身のガイは
イスラエル軍の
サイバーセキュリティ部門を経て
2002年
アプリケーションセキュリティの
パイオニアである
サンクタムに入社しました


開発プロセスの早い段階で
アプリスキャンを行って
問題を発見することが重要だよなぁ…
修正も早く行うことで
リリースを早めることにも
開発費用を抑えることにも繋がるし

2004年
サンクタムは
アプリケーションセキュリティの会社である
ウォッチファイヤーに買収されました

2004

watchfire

ウォッチファイヤーの拠点は
カナダのオタワ

そこに
プロジェクトマネージャーとして一名
サンクタムから派遣するように
依頼しました

これは
大チャンスだ！

そう意気込んで
イスラエルにいたガイは
25歳にしてカナダへと
向かったのでした

さ　さむい…

ヒュウウウウウ…

ひとつのプロジェクトの
マネージャーという立ち位置で
ウォッチファイヤーに来たガイは
まだまだ無名の一般社員でした

しかし
やる気に満ちあふれて働き
様々な社員との懸け橋としても
活躍したガイは
一躍スター社員となったのです

のちに最高の
ビジネスパートナーとなる
ピーターとの出会いもここでした

当時
ウォッチファイヤーの
CEOだったピーターは
最初こそガイの存在を
知らなかったものの
その躍進で知ることとなったのです

その後
ウォッチファイヤーは
IBMに買収され…

ガイは退社ののち
ウォッチファイヤーの
CTOだったマイクと一緒に
ブレイズという会社を設立しました





BLAZE
INTELLIGENT OPTIMIZATION

ブレイズは二年程度で
アカマイに買収され



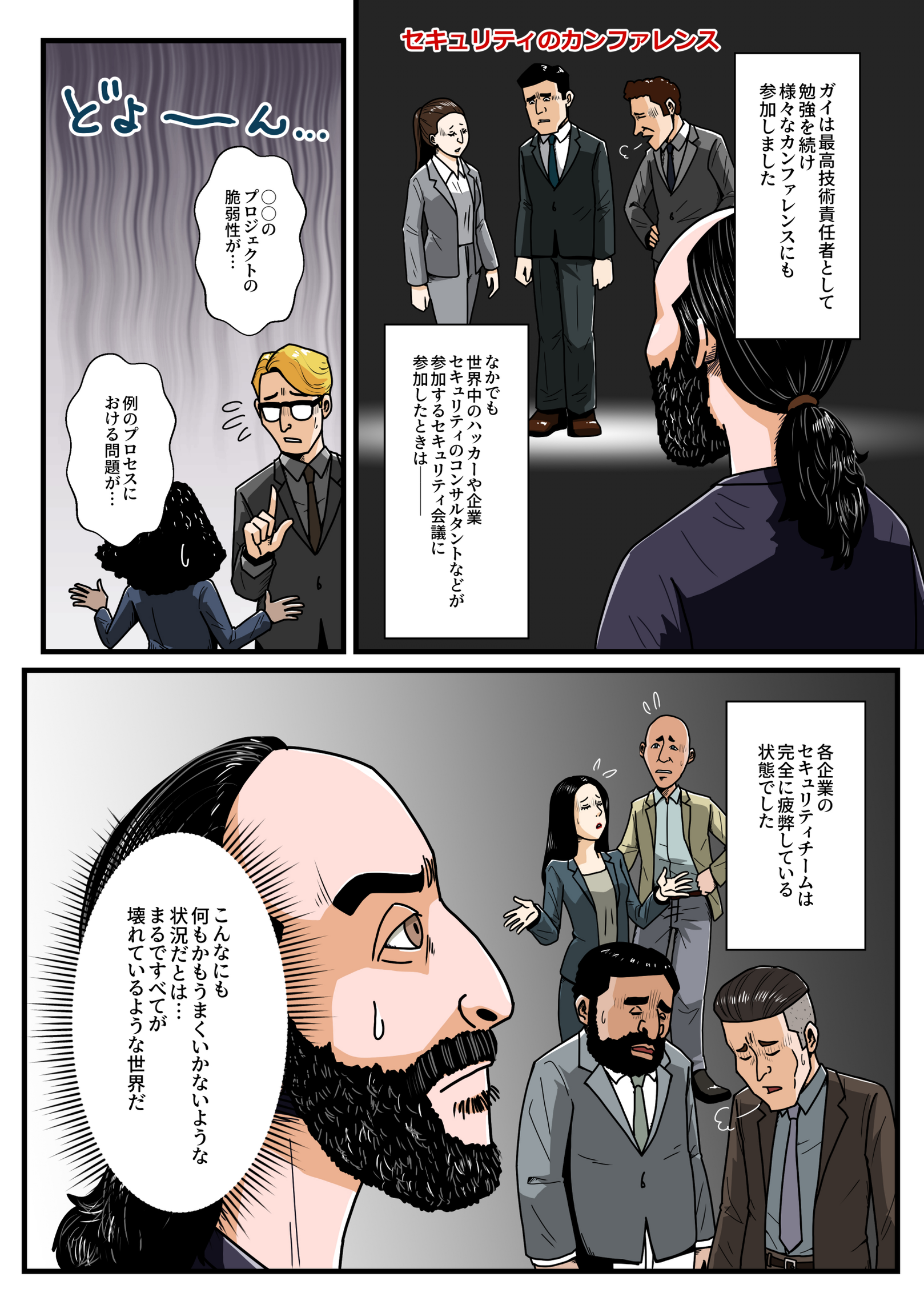
ブレイズは
Webパフォーマンスオプティマイゼーションの会社で
ウォッチファイヤーでガイの活躍を見たピーターは
ブレイズの投資家であり
役員としてもガイを支えました



ガイは開発者とセキュリティチーム
それぞれが円滑に進められるための構想を
より具体化できるよう
試行錯誤を重ねていきました


うーん…
どうすれば良いのか…
最後の段階でチェックする
セキュリティチームの
負担を減らさないとだよなぁ…
…よし！
セキュリティチームが
開発段階でチェックできるよう
監査ツールを開発して導入しよう！
ピーターの支えもあり
ガイの計画は
着々と進んだように見えたが……
セキュリティ担当の
チェックがまだ終わりません！
このままじゃ
リリースの早期化どころか
どんどん後ろ倒しに…！
ここの脆弱性も…
ここも
問題じゃないか？
ええ!?


セキュリティチームが
途中でチェックしてアドバイスできれば
最後にまとめてチェックを行う
負担も少なくなって
開発側も最初からやり直す
手間がなくなると思ったのに…！
セキュリティチームが
開発現場の状況がわからないまま
フィードバックできる環境を作ったことで
かえって現場が混乱し
今までの4倍もの時間を要する結果に…
開発A
セキュリティチームは
口を出すだけだから
なんとでも言えるんだよな！
こんな状態じゃ
今までのフローの大変さと
ちっとも変わらない！
セキュリティA
まずい
このままではいけない…


セキュリティのカンファレンス
ガイは最高技術責任者として
勉強を続け
様々なカンファレンスにも
参加しました
なかでも
世界中のハッカーや企業
セキュリティのコンサルタントなどが
参加するセキュリティ会議に
参加したときは――
どよーーん…
○○の
プロジェクトの
脆弱性が…
例のプロセスに
おける問題が…
各企業の
セキュリティチームは
完全に疲弊している
状態でした
こんなにも
何もかもうまくいかないような
状況だとは…
まるですべてが
壊れているような世界だ

一方
DevOpsに関する
カンファレンスに参加すると

♫わい
わい♫

みんなハレルヤを
歌っているような
明るく、各チームが
手を取り合って笑っているような
雰囲気だったのです

私が求めているのは
何もかもが壊れているような
状況ではなく
こういう世界なのに……

こうしてガイの中で
長年構想されていた
開発・セキュリティの現状を
変えていかなければ
ならないという想いは
どんどん強くなっていきました

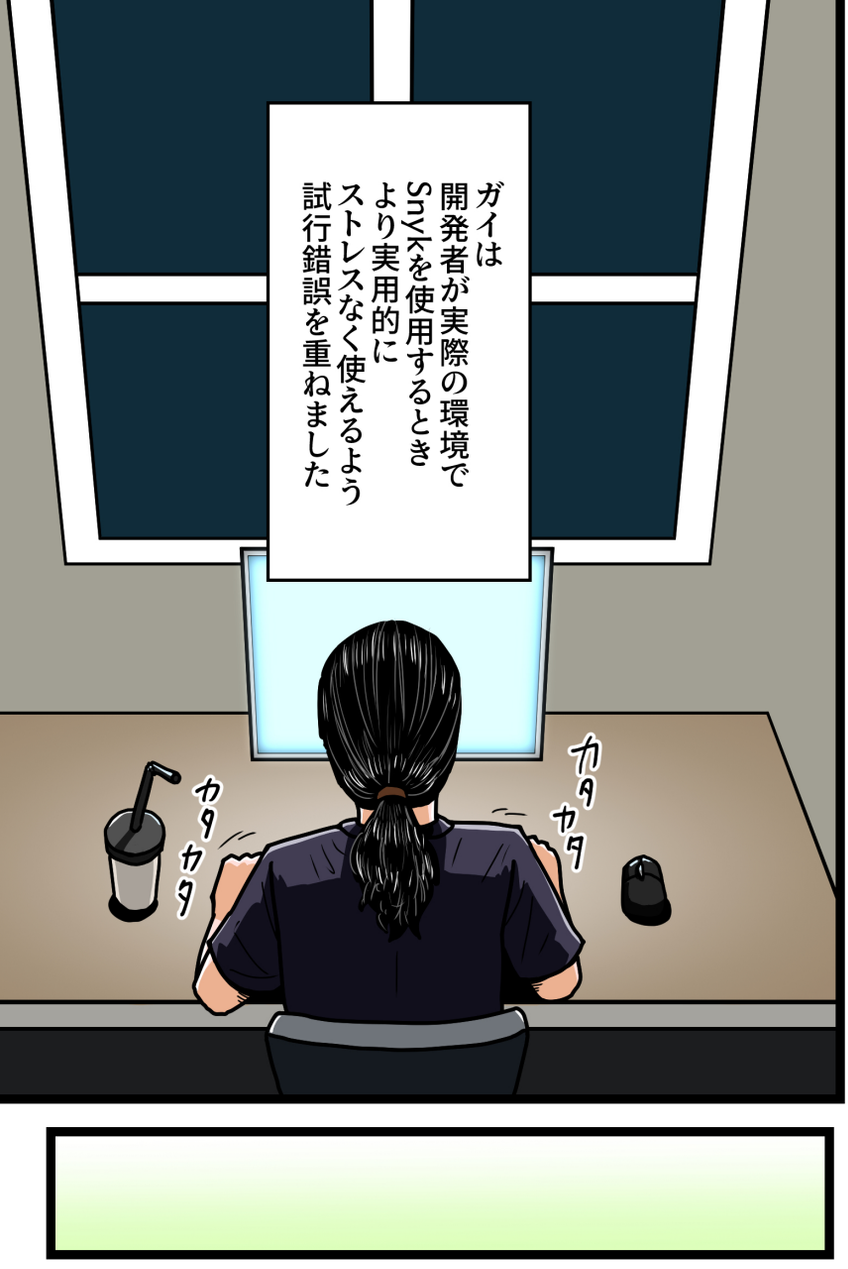


開発者も
セキュリティチームも
笑顔で手を取り合えるような

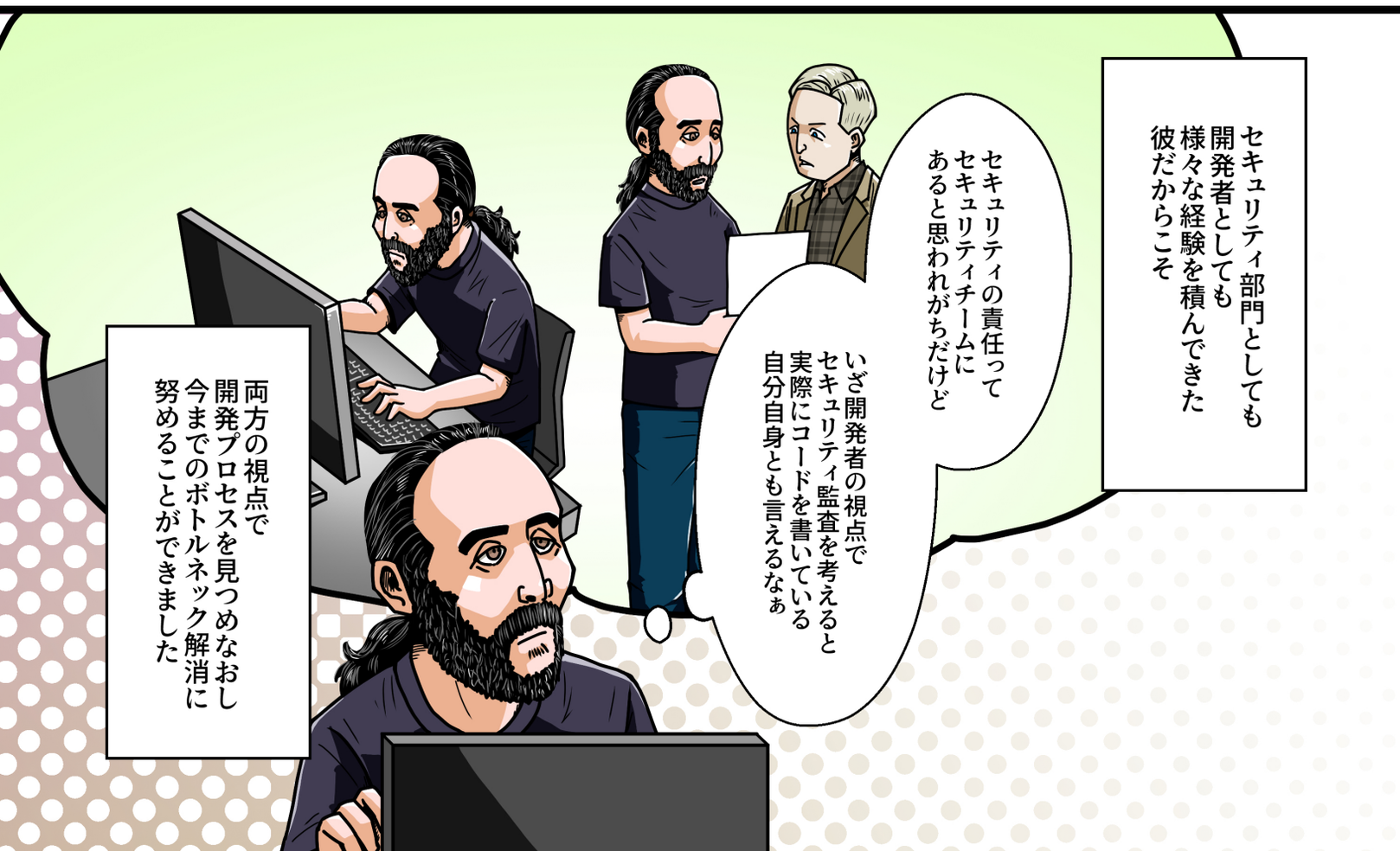
開発者のための
セキュリティソリューションを
構築しなければならない

そこで
生まれたのが
Snykでした

snyk


ガイは
開発者が実際の環境で
Snykを使用するとき
より実用的に
ストレスなく使えるよう
試行錯誤を重ねました
カタカタ
カタカタ

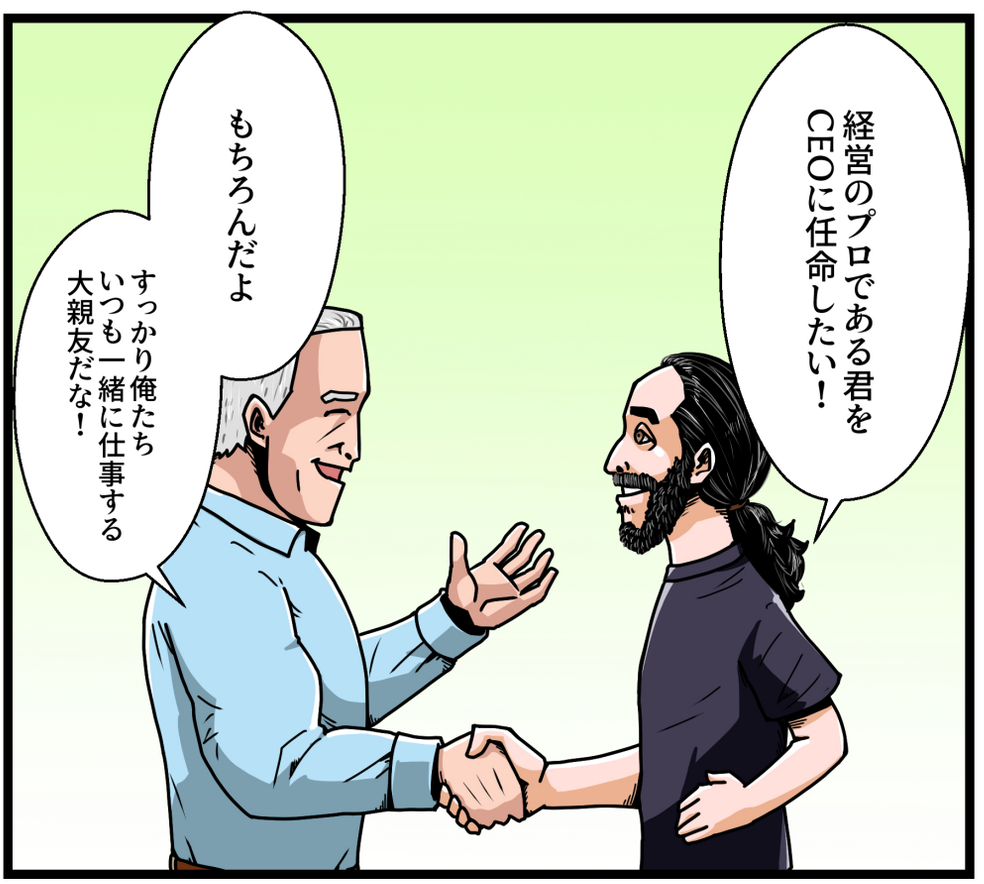

開発しながら
リアルタイムで
セキュリティのチェックが
できることは大前提として…
開発者の負担を
減らすために
修正のヒントも
同時に出せるように
しなければいけないな

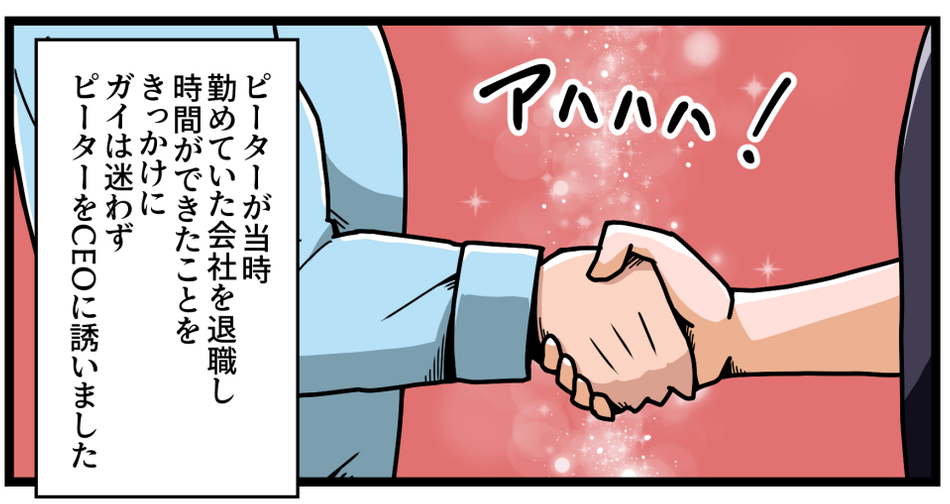

セキュリティ部門としても
開発者としても
様々な経験を積んできた
彼だからこそ
セキュリティの責任って
セキュリティチームに
あると思われがちだけど
いざ開発者の視点で
セキュリティ監査を考えると
実際にコードを書いている
自分自身とも言えるなぁ
両方の視点で
開発プロセスを見つめなおし
今までのボトルネック解消に
努めることができました



もちろん
かつてより言われていた
シフトレフトの
動きについても……

こうしてSnykは
DevSecOpsに向けて
開発が進められていき・・・





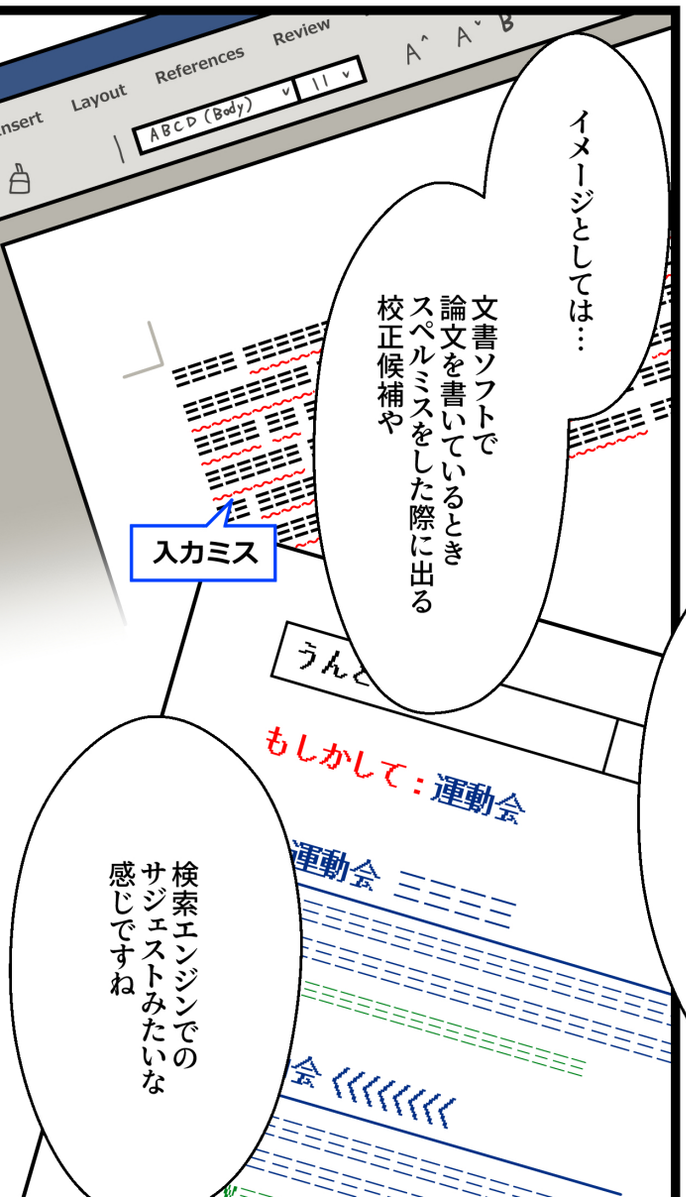
ピーターが当時
勤めていた会社を退職し
時間ができたことを
きっかけに
ガイは迷わず
ピーターをCEOに誘いました

実際にSnykは
どのような形で
開発されていったのか
というと…

新しいツールを
組み込むにあたって
まずは実際にコードを書く
開発者が違和感なく
使えなければならない

だから
新たにアプリを
一から使い慣らしたり
準備する手間がないよう

普段使っているツールに
組み込めるような
システムにしました



イメージとしては…

文書ソフトで
論文を書いているとき
スペルミスをした際に出る
校正候補や

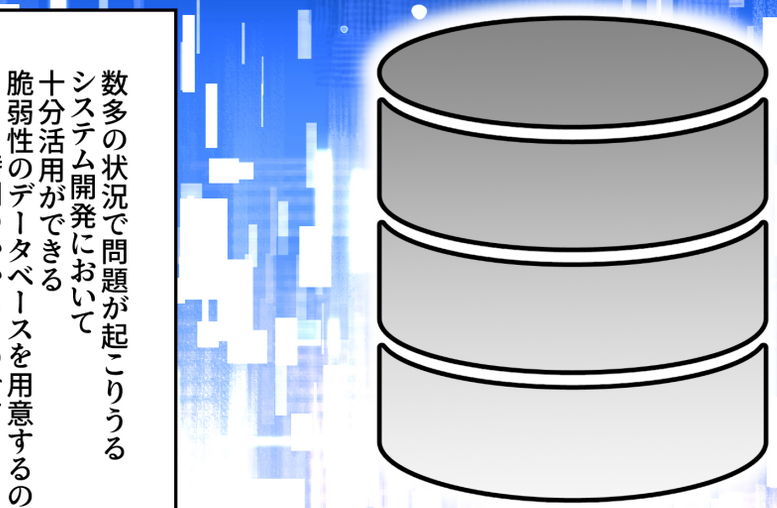
検索エンジンでの
サジェストみたいな
感じですね

次に
セキュリティチームからの指摘と
開発者のすれ違いから
学んで…

脆弱性に関する問題を
指摘するだけでなく
修正のためのアドバイスを
出せるようにしました

ただ
これらを現実にするには
脆弱性に関する
膨大な量のデータベースが
必要でした

数多の状況で問題が起こりうる
システム開発において
十分活用ができる
脆弱性のデータベースを用意するのは
とても時間のかかるものです



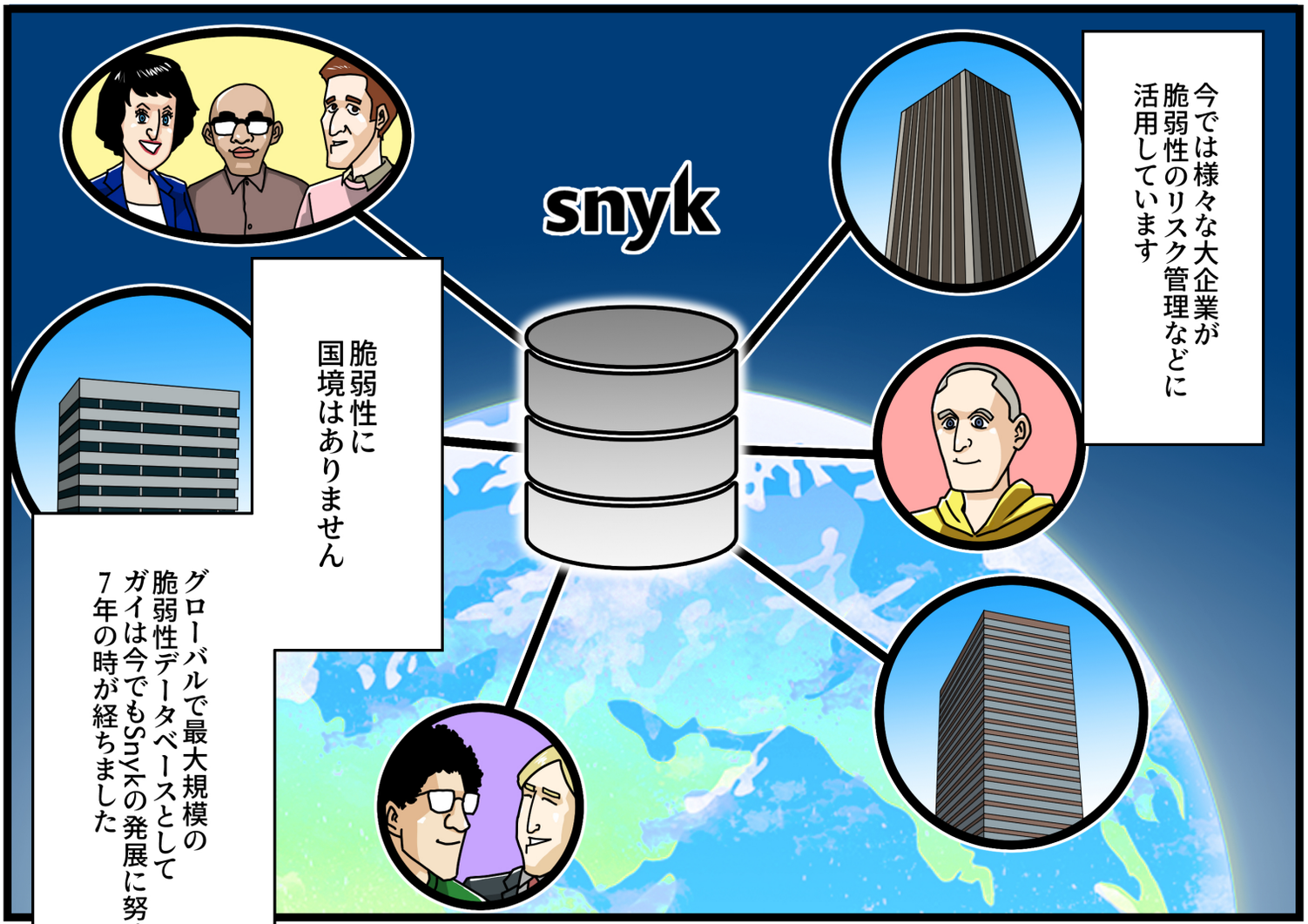
ガイは世界中の
セキュリティ第一人者や
専門チームから
知見を集めたり

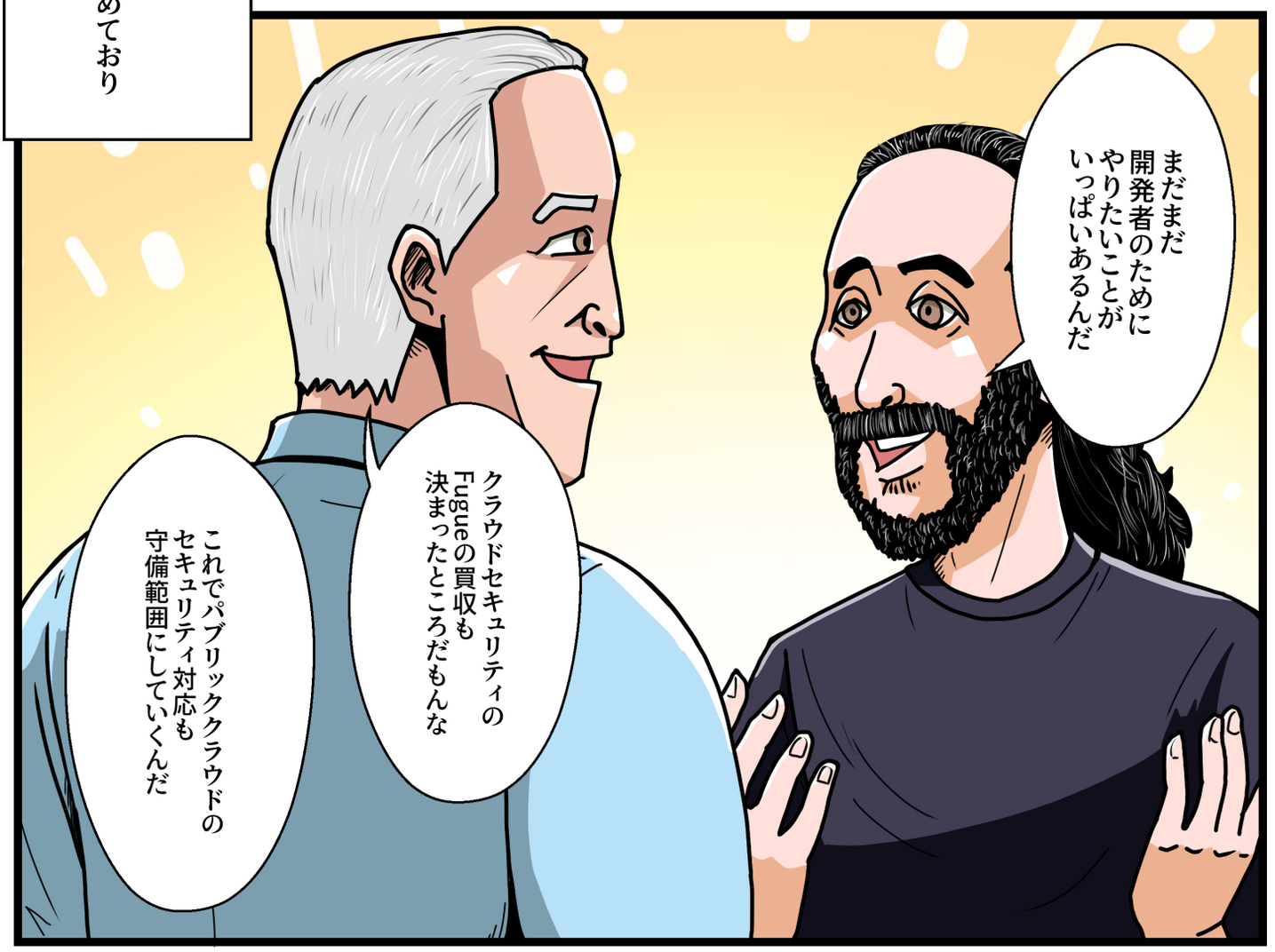
SECURITY
専門チーム

バグに報酬金をかけて
一般人に見つけてもらう
バグバウンティを導入するなど
様々な方法で
脆弱性に関するデータを
集めていったのです

当初は
開発者とセキュリティチームのために
集めたデータベースは

多くのデベロッパーコミュニティや
企業、行政機関などにおいて
サイバー攻撃を防ぐために
活用されるようになりました

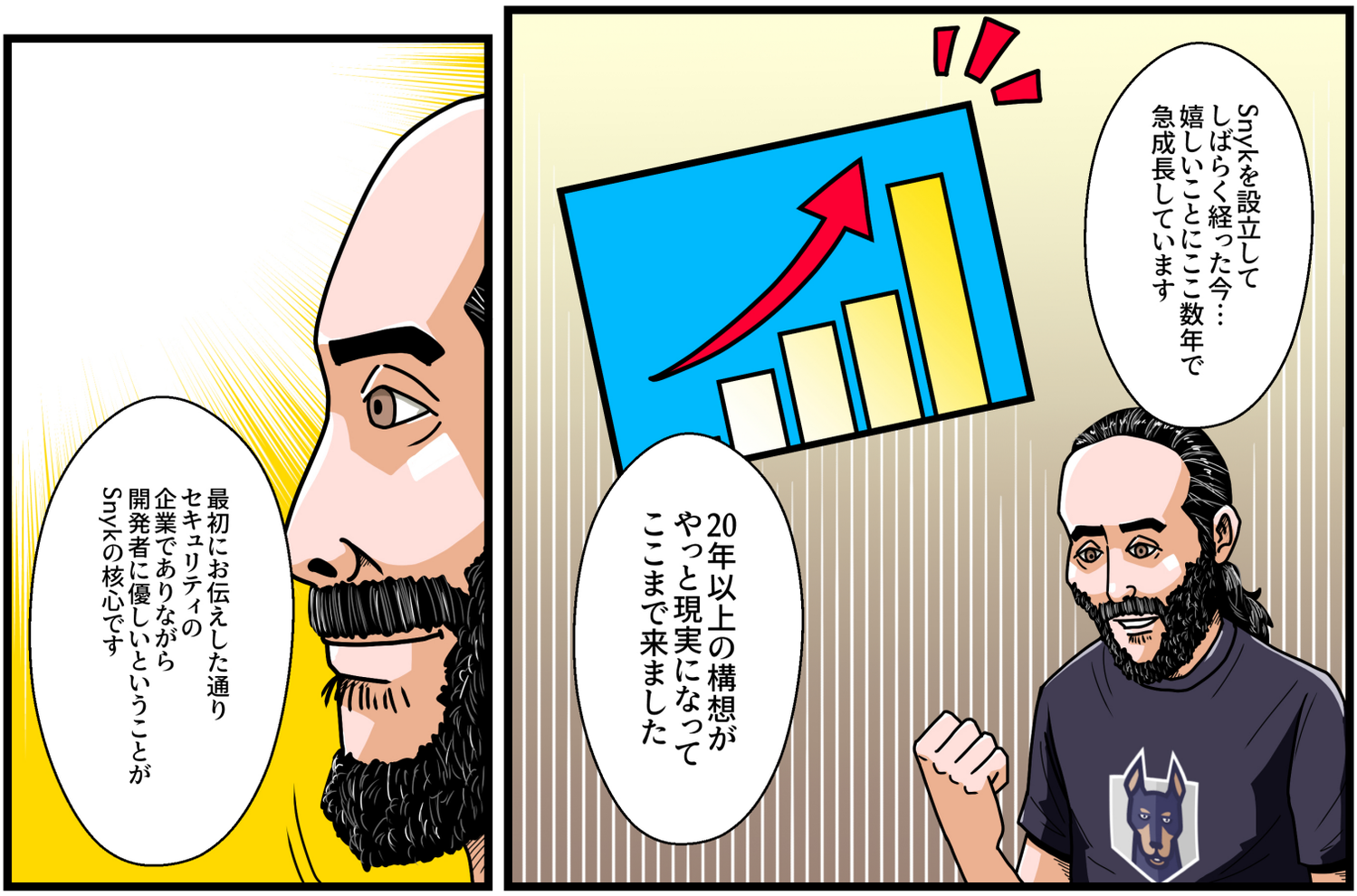

今では様々な大企業が
脆弱性のリスク管理などに
活用しています
snyk
脆弱性に
国境はありません
グローバルで最大規模の
脆弱性データベースとして
ガイは今でもSnykの発展に努めており
7年の時が経ちました

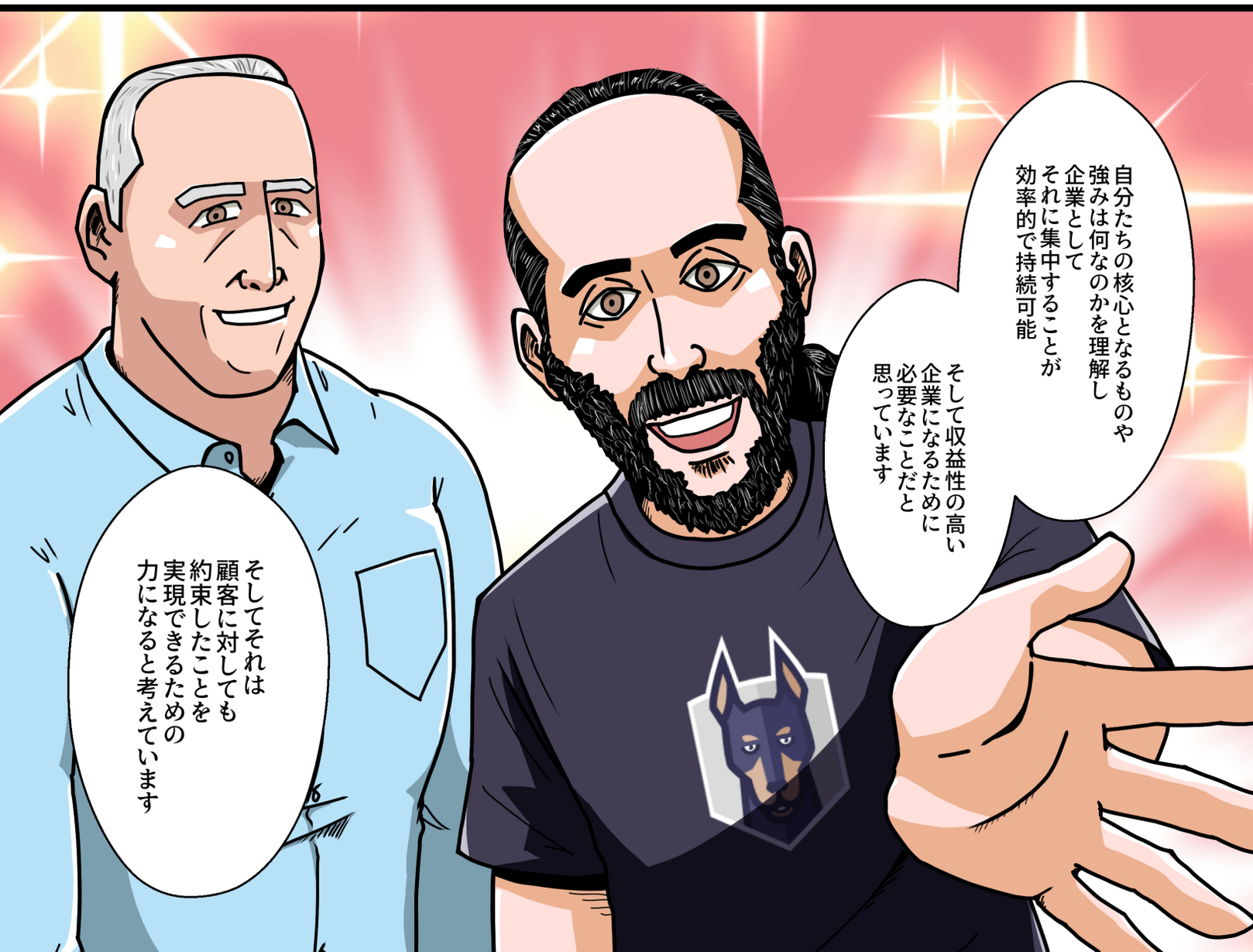

まだまだ
開発者のために
やりたいことが
いっぱいあるんだ
クラウドセキュリティの
Fugueの買収も
決まったところだもんな
これでパブリッククラウドの
セキュリティ対応も
守備範囲にしていくんだ


開発とセキュ
両立
一つ目は開発中
つまり
迅速な開発と
セキュリティを
両立するということです
セキュリティと
開発に対する
取り組み方は
大きく分けて
三つに分けられます
Snykは
それらに対応する
ツールとして
活用できるのです
通常
開発が終わった後に
セキュリティチェックを
行うのがこれまでの流れでした
そのとき
脆弱性が見つかると
セキュリティチームが
開発者に修正の指示を出し
そのあとに開発者が
一生懸命修正する…
こうしてリリースが
遅れてしまいます
しかし
Snykを活用すれば
開発者がコードを書きながら
脆弱性を修正できるので
開発完了後すぐに
リリースができるのです


二つ目は
セキュリティを
品質の一部として
捉えることです
定期的に
外部のベンダーへ
脆弱性の診断や
セキュリティ検査を
依頼する企業も多いです
しかしコードは
都度更新される上に
診断だけでは
修正方法がわからない
こともあります
Snykの
リアルタイムで
問題を確認できる
仕様は
セキュリティの専門家でなくても
常に安全な品質を保てる上に
社内の開発者に
ノウハウが蓄積していくのです
三つ目は
新しい脆弱性への
レスポンスです
HeartbleedやLog4Shellなど
重大な新しい脆弱性は
いつ発見されるか
誰にもわかりません
Snykを活用することで
新しい脆弱性が
発見されても
すぐに影響範囲を把握して
修正できる…
もう気合と根性で
脆弱性を
管理する時代は
終わったのです！


つまり
実質的に
セキュリティチームが
いなくても使用できる
Snykですが
決してプロセスから
セキュリティチームが
必要なくなるわけでは
ありません
開発者自身の
セキュリティ意識を
高めながら
セキュリティチームには
これまでの知見が必要な
重要で複雑な場面に
直接関わってもらうことに
なります
開発者
セキュリティ
担当
数々の起業や
DevOpsへの関わりを
通してみてきた
開発者とセキュリティチームの
関係性や
セキュリティチェックで
発生してしまうボトルネック…
これらの解消を
ずっと理想として
掲げてきました
それが
Snykのツールを
活用することで
実現するのです！


Snykを設立して
しばらく経った今…
嬉しいことにここ数年で
急成長しています
20年以上の構想が
やっと現実になって
ここまで来ました
最初にお伝えした通り
セキュリティの
企業でありながら
開発者に優しいということが
Snykの核心です


自分たちの核心となるものや
強みは何なのかを理解し
企業として
それに集中することが
効率的で持続可能
そして収益性の高い
企業になるために
必要なことだと
思っています
そしてそれは
顧客に対しても
約束したことを
実現できるための
力になると考えています


私たちが
顧客に
約束することは
開発者が実際に
私たちの製品を使って
その開発者が触れる
すべてのものを
私たちが安全にする
ということです
snyk
開発者セキュリティに
必要なことは
まだまだ揃っては
いません
わっ！
そのために
私たちはこれからも
前進し続けます



snyk

マンガでわかる!

Snyk誕生物語

Peter McKay and Guy Podjarny