



Keyword-Driven-Testフレームワークによる 自動システムテストの実現

株式会社エビデント

SW Microscope and Engineering : 江良 徹

Agenda

- 1 自己紹介、2022年の発表について
- 2 新たな自動テストの課題
- 3 Keyword-Driven-Testフレームワークの説明
- 4 メリット
- 5 エ夫したこと
- 6 まとめ

はじめに：開発製品の紹介



ライフサイエンスソリューション

OEMソリューション

産業ソリューション

ライフサイエンスリサーチ



細胞培養モニタリング



工業用顕微鏡



工業用ビデオスコープ、ボアスコープ



クリニカル



教育・実習



非破壊検査機器



蛍光X線分析装置



ハードウェアを制御するソフトウェアがターゲット

2022年ソフトウェア品質シンポジウムにおいて

ソフトウェアテスト自動化を進める中で出た問題について発表した
今回はその内容に対するアップデートの発表となる


A3-1 経験発表

組織的にシステムテスト自動化を推進する体制の構築

登壇者
内山 守 氏
株式会社エビデント

共著
株式会社エビデント
江良 徹 氏 石野 武 氏

昨今、ソフトウェア開発サイクルが短くなり、ソフトウェアテストにおいても、スピード、正確性、および費用対効果を求められている。弊社では、従来の手動テストだけでは品質の維持および時間効率に課題があるため、システムテストの自動化を進めている。



しかしテスト自動化は製品個別で推進されるため、担当者の力量に左右される傾向が大きい。そのため、①テスト自動化の活動が継続しない、②重複した検討作業を行い非効率、③自動化の推進に消極的、といった問題を抱えていた。

このような属人的な状況を打破するためには、組織的にテスト自動化に取り組む必要があると考えた。まずは、部門としてテスト自動化戦略を作成し、組織としてのテスト自動化の実現を目標として掲げた。加えて、テスト自動化の普及および定着を実現するための実行部隊としてWorking Groupを結成し活動を推進した。

2022年発表資料より

自動テストへの移行を推進

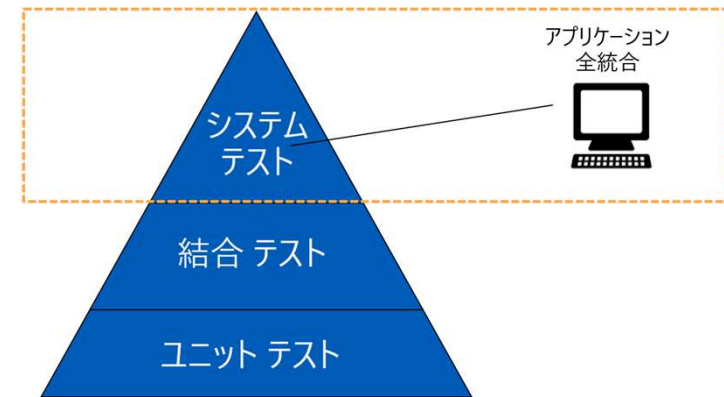
これまで、システムテストは手動テストを中心



効率化の視点で自動テストへのシフトを推進

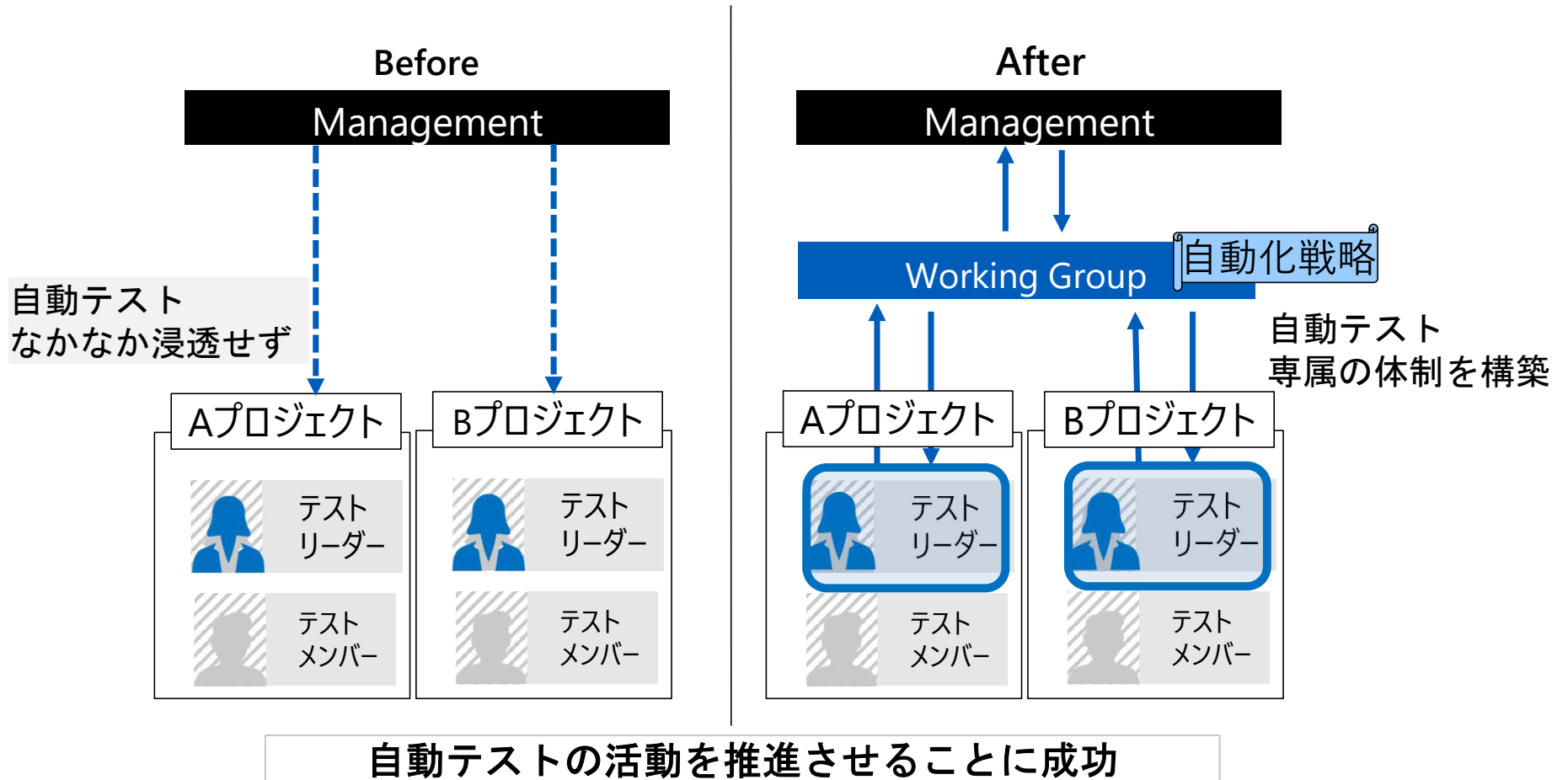
【2つの対策を実施】

- ①自動テストを進めるにあたっての体制を構築し、テスト戦略を設定
- ②製品毎での自動テスト検討をやめ、知見を共通化



2022年発表資料より

① 自動テストを進めるにあたっての体制を構築し、テスト戦略を設定



Agenda

1 自己紹介、2022年の発表について

2 **新たな自動テストの課題**

3 Keyword-Driven-Testフレームワークの説明

4 メリット

5 エ夫したこと

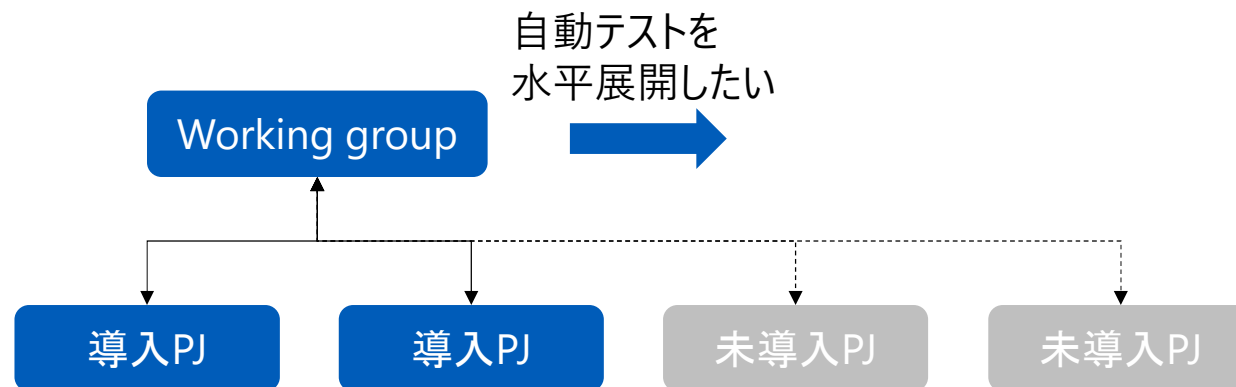
6 まとめ

新たな取り組み

新たに、自動テスト未導入のプロジェクトにも水平展開したい

これまでは、ある程度自動テストの知識・経験あるプロジェクトを対象にしていた

水平展開の場合、自動テストの知識・経験のないプロジェクトも対象にする必要性



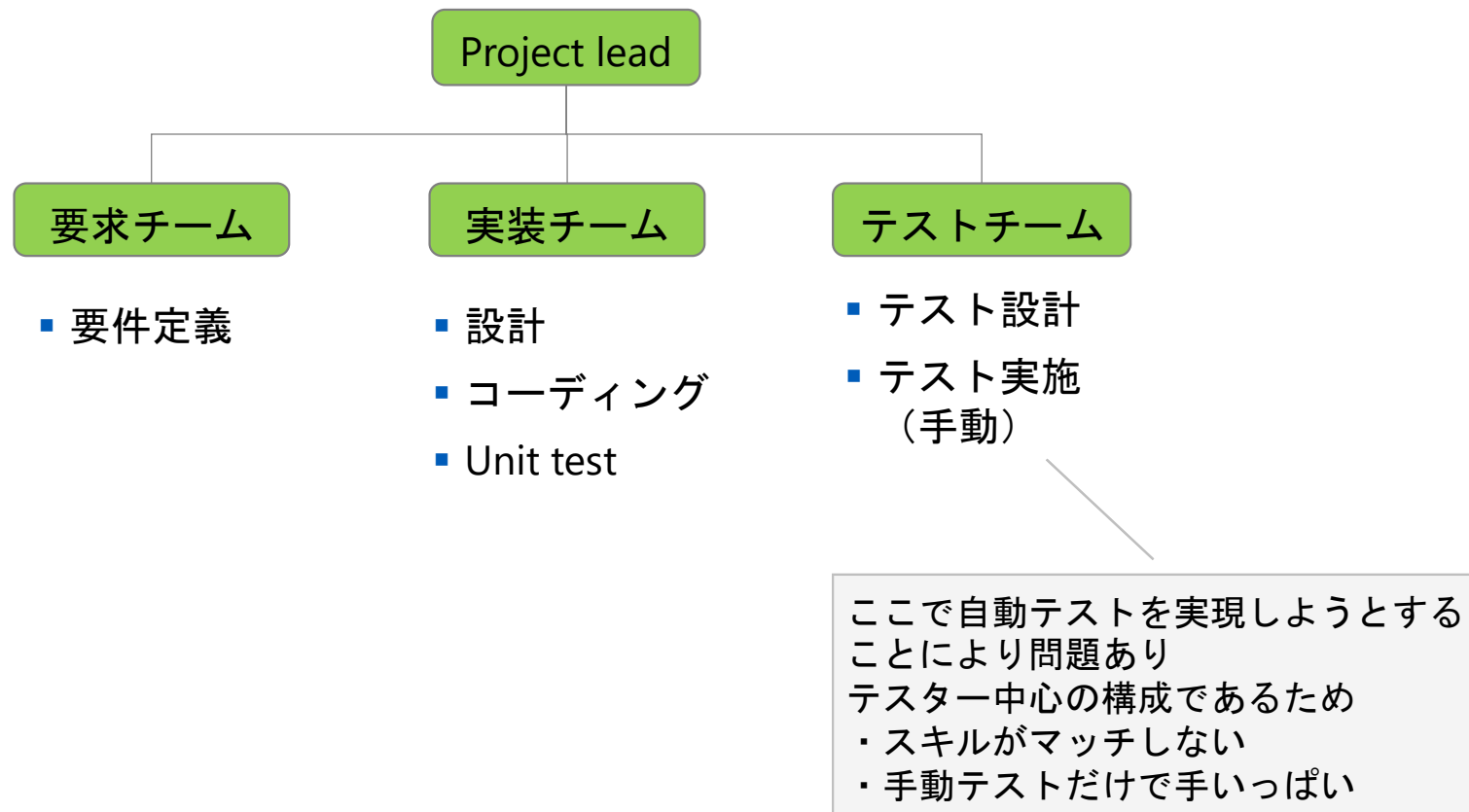
水平展開時に出た問題

新規導入するとき、問題が出て導入が進まないケースあり

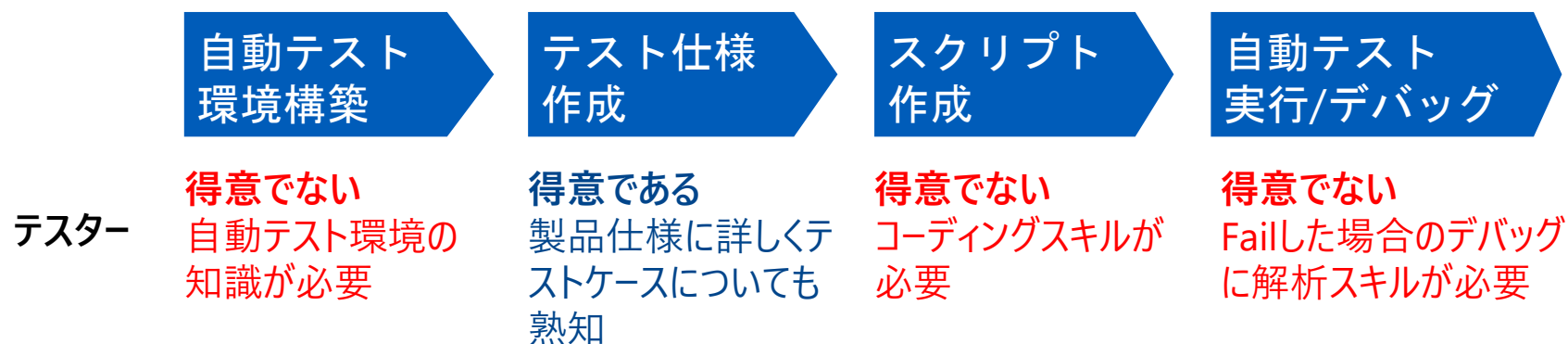
【自動テスト導入にあたっての問題】

- ①テストにアサインされたテスターでは、自動テストを構築できない
- ②自動テストの共通ノウハウはあるが、それをプロジェクトに落とし込めない

問題の背景：一般的な開発体制



問題の背景：一般的な自動テストのプロセス



自動テストを実現するには、自動テスト特有のエンジニアリングスキルも必要
そこがハードルになっている

課題設定

課題 1 : 自動テストの役割を、**エンジニアリング部分**と**テスト設計部分**に分割して活動できるよう、ツールを開発

「Keyword-Driven-Testフレームワーク」

課題 2 : テストチームで不足している**エンジニアリング部分**の支援体制を構築

Agenda

- 1 自己紹介、2022年の発表について
- 2 新たな自動テストの課題
- 3 Keyword-Driven-Testフレームワークの説明**
- 4 メリット
- 5 エ夫したこと
- 6 まとめ

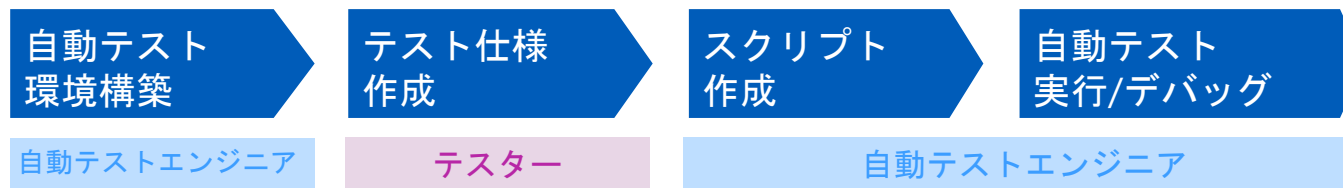
Keyword-Driven-Testフレームワーク（KDT）のコンセプト

【コンセプト】

- 自動テストの役割を、**エンジニアリング部分**と**テスト設計部分**に分割
- **エンジニアリング部分**は自動テストエンジニアが担当
- **テスト設計部分**はテスターが担当。テスターはKeywordを選択しExcelでテスト仕様書を作成
- この構成をフレームワーク化することで、他製品でも活用

KDTにおける役割分割

テスター、自動テストエンジニアの得意領域に専念できるよう、役割分担



テスター

自動テストのテスト仕様作成に注力

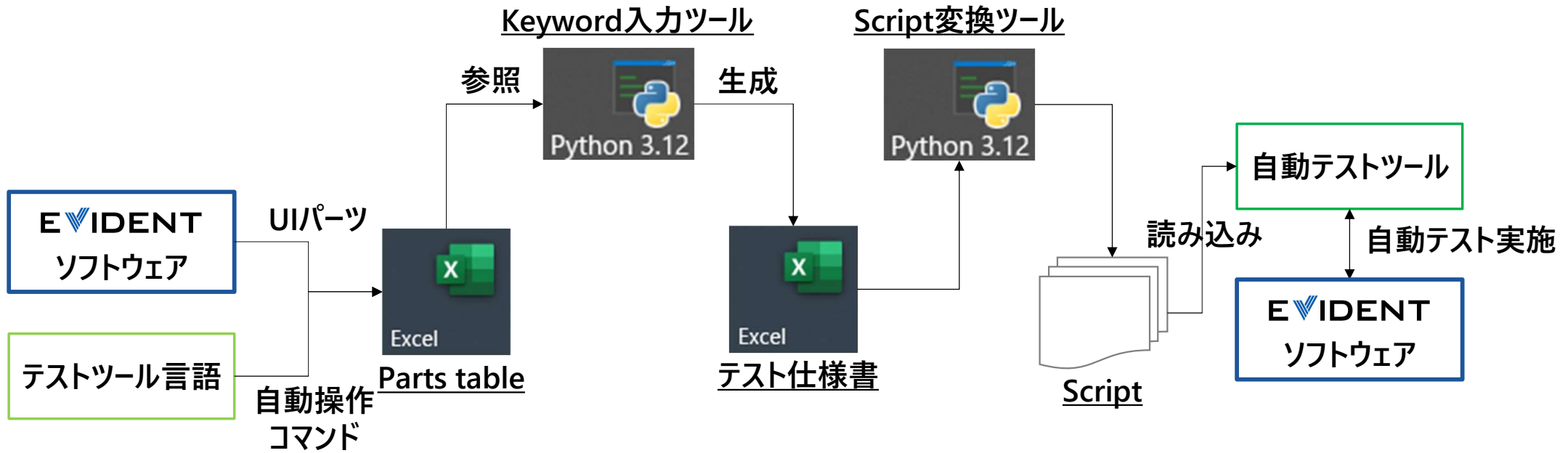
- ・自動テストフレームワークを利用し、自動テスト仕様書の作成（Excel）

自動テストエンジニア

自動テストのエンジニアリング部分に注力

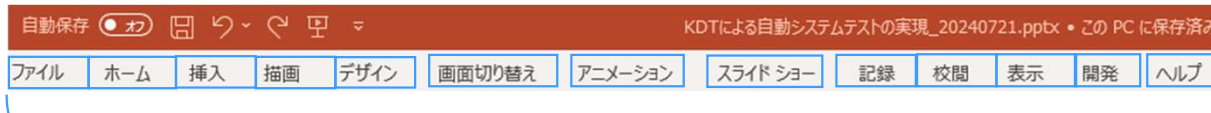
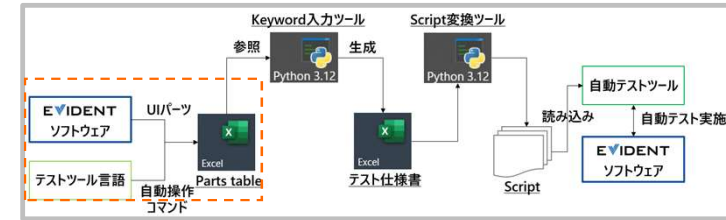
- ・自動テストフレームワークの構築
- ・自動テストのデバッグ

KDTフレームワーク アーキテクチャ



自動テストツール は市販ソフトを買い入れ

Parts table (Microsoft PowerPointの例)



Parts table

IDラベル Parts ID	単純キーワード (Command Level) Keyword	アクション Action	imageName
Parts_Button_Sanshou(GUI_FileOp	[GUI]FileOpen > [Button]Sanshou	ACTION_UI_MOUSE_CLICK	GUI_FileOpen/Button_Sanshou
Parts_Label_FileOpenDialog(GUI_f	[GUI]FileOpen > [Label]FileOpenDialog	ACTION_UI_MOUSE_CLICK	GUI_FileOpen/Label_FileOpenDialog
Parts_Menu_FileOpen(GUI_FileOp	[GUI]FileOpen > [Menu]FileOpen	ACTION_UI_MOUSE_CLICK	GUI_FileOpen/Menu_FileOpen
Parts_Menu_02File(GUI_Menu)	[GUI]Menu > [Menu]02File	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_02File
Parts_Menu_03Home(GUI_Menu)	[GUI]Menu > [Menu]03Home	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_03Home
Parts_Menu_04Insert(GUI_Menu)	[GUI]Menu > [Menu]04Insert	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_04Insert
Parts_Menu_05Draw(GUI_Menu)	[GUI]Menu > [Menu]05Draw	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_05Draw
Parts_Menu_06Design(GUI_Menu)	[GUI]Menu > [Menu]06Design	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_06Design
Parts_Menu_07ImageChange(GUI	[GUI]Menu > [Menu]07ImageChange	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_07ImageChange
Parts_Menu_08Animation(GUI_Me	[GUI]Menu > [Menu]08Animation	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_08Animation
Parts_Menu_09SlideShow(GUI_Me	[GUI]Menu > [Menu]09SlideShow	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_09SlideShow
Parts_Menu_10Record(GUI_Menu)	[GUI]Menu > [Menu]10Record	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_10Record
Parts_Menu_11Check(GUI_Menu)	[GUI]Menu > [Menu]11Check	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_11Check
Parts_Menu_12Display(GUI_Menu)	[GUI]Menu > [Menu]12Display	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_12Display
Parts_Menu_13Develop(GUI_Men	[GUI]Menu > [Menu]13Develop	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_13Develop
Parts_Menu_14Help(GUI_Menu)	[GUI]Menu > [Menu]14Help	ACTION_UI_MOUSE_CLICK	GUI_Menu/Menu_14Help

自動操作
コマンド

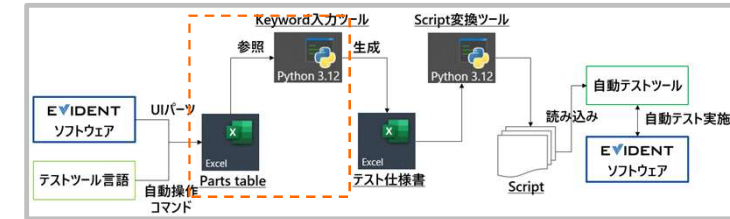
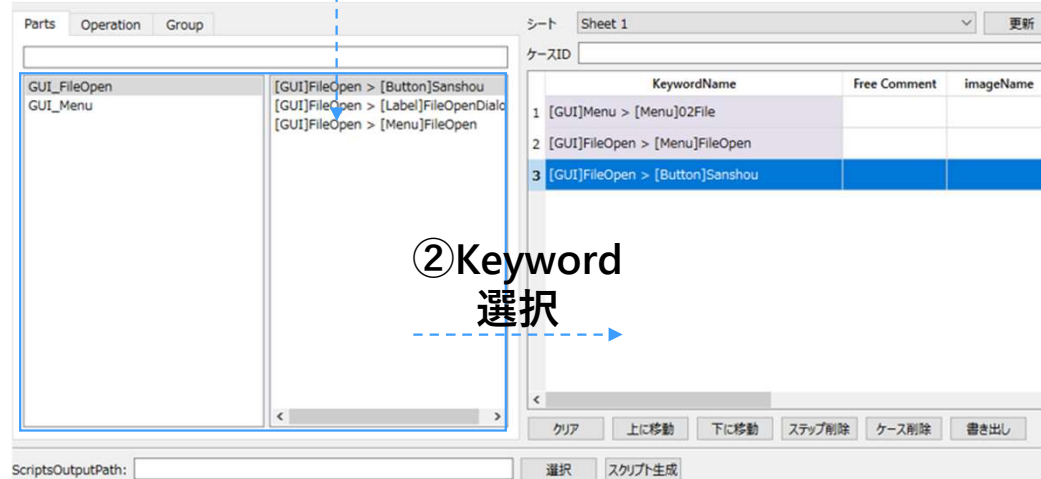
Keyword入カツール

Parts table

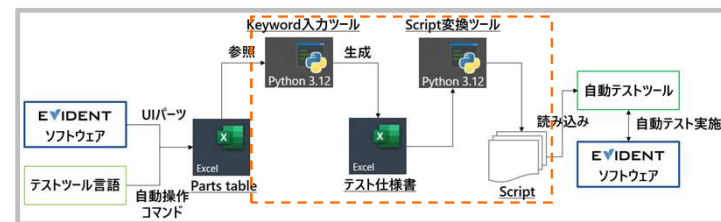
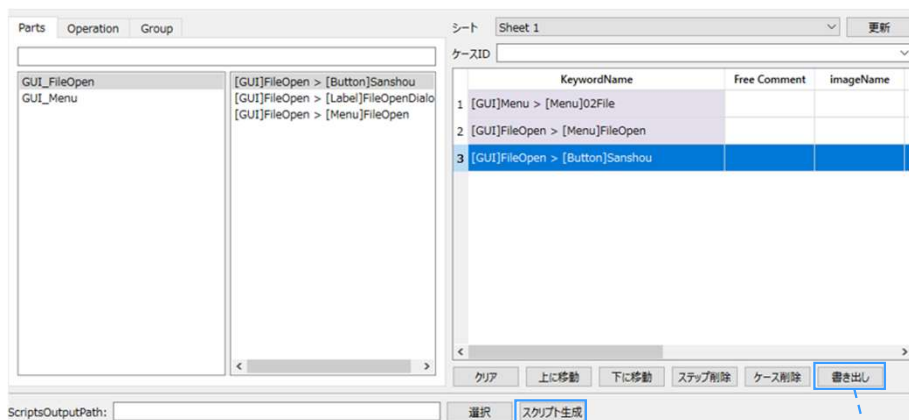
IDラベル Parts ID	単純キーワード (Command Level) Keyword	アクション Action	imageName
Parts_Button_Sanshou(GUI_FileOp	[GUI]FileOpen > [Button]Sanshou	ACTION_UI_MOUSE_CLICK GUI_FileOpen/Button_Sanshou	
Parts_Label_FileOpenDialog(GUI_f	[GUI]FileOpen > [Label]FileOpenDialog	ACTION_UI_MOUSE_CLICK GUI_FileOpen/Label_FileOpenDialog	
Parts_Menu_FileOpen(GUI_FileOp	[GUI]FileOpen > [Menu]FileOpen	ACTION_UI_MOUSE_CLICK GUI_FileOpen/Menu_FileOpen	
Parts_Menu_02File(GUI_Menu)	[GUI]Menu > [Menu]02File	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_02File	
Parts_Menu_03Home(GUI_Menu)	[GUI]Menu > [Menu]03Home	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_03Home	
Parts_Menu_04Insert(GUI_Menu)	[GUI]Menu > [Menu]04Insert	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_04Insert	
Parts_Menu_05Draw(GUI_Menu)	[GUI]Menu > [Menu]05Draw	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_05Draw	
Parts_Menu_06Design(GUI_Menu)	[GUI]Menu > [Menu]06Design	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_06Design	
Parts_Menu_07ImageChange(GUI	[GUI]Menu > [Menu]07ImageChange	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_07ImageChange	
Parts_Menu_08Animation(GUI_Me	[GUI]Menu > [Menu]08Animation	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_08Animation	
Parts_Menu_09SlideShow(GUI_Me	[GUI]Menu > [Menu]09SlideShow	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_09SlideShow	
Parts_Menu_10Record(GUI_Menu)	[GUI]Menu > [Menu]10Record	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_10Record	
Parts_Menu_11Check(GUI_Menu)	[GUI]Menu > [Menu]11Check	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_11Check	
Parts_Menu_12Display(GUI_Menu)	[GUI]Menu > [Menu]12Display	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_12Display	
Parts_Menu_13Develop(GUI_Men	[GUI]Menu > [Menu]13Develop	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_13Develop	
Parts_Menu_14Help(GUI_Menu)	[GUI]Menu > [Menu]14Help	ACTION_UI_MOUSE_CLICK GUI_Menu/Menu_14Help	

Keyword入カツール

①参照

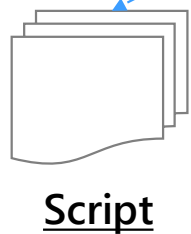


テスト仕様書、Script出力

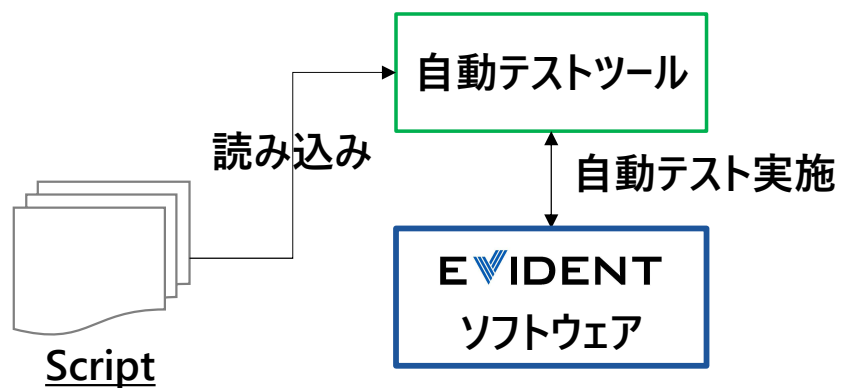
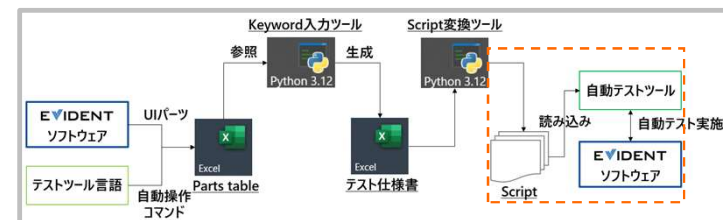


Excel出力

Manual Test Case ID	Excel出力		
	No.	Keyword Name	Category
0001_SampleTest	1	[GUI]Menu > [Menu]02File	Step
	2	[GUI]FileOpen > [Menu]FileOpen	Step
	3	[GUI]FileOpen > [Button]Sanshou	Step

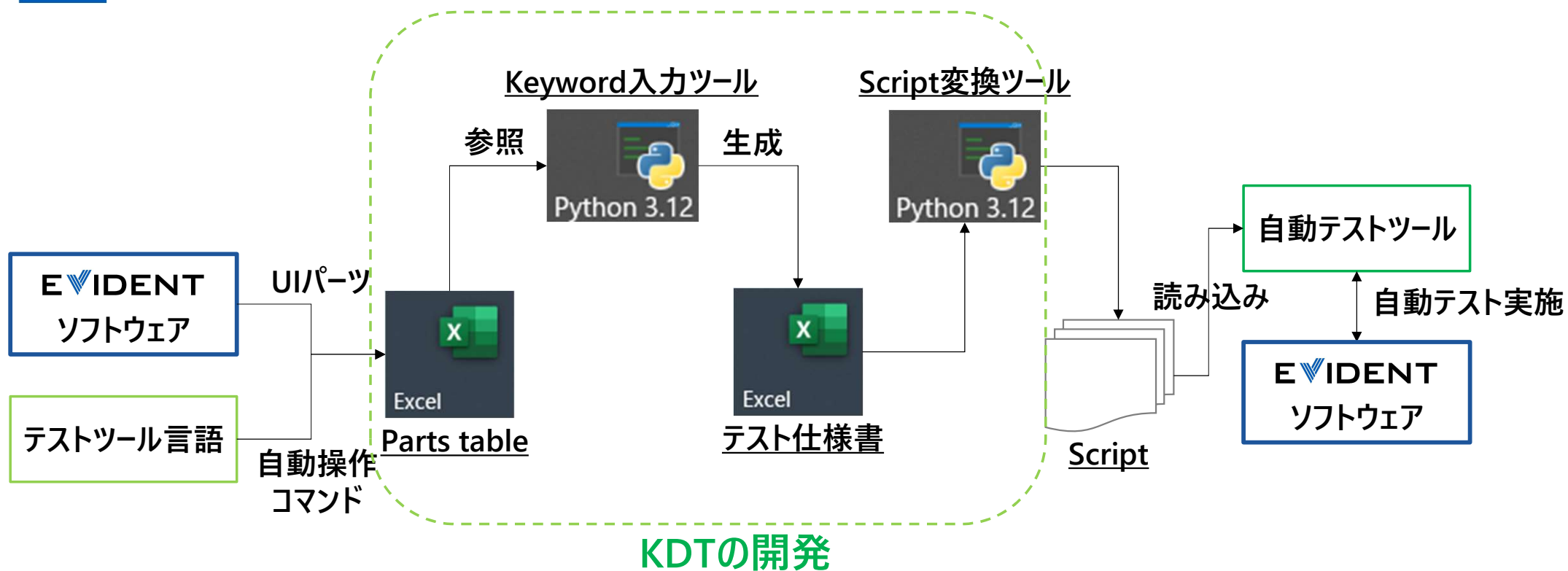


Scriptを使ってテストを実行

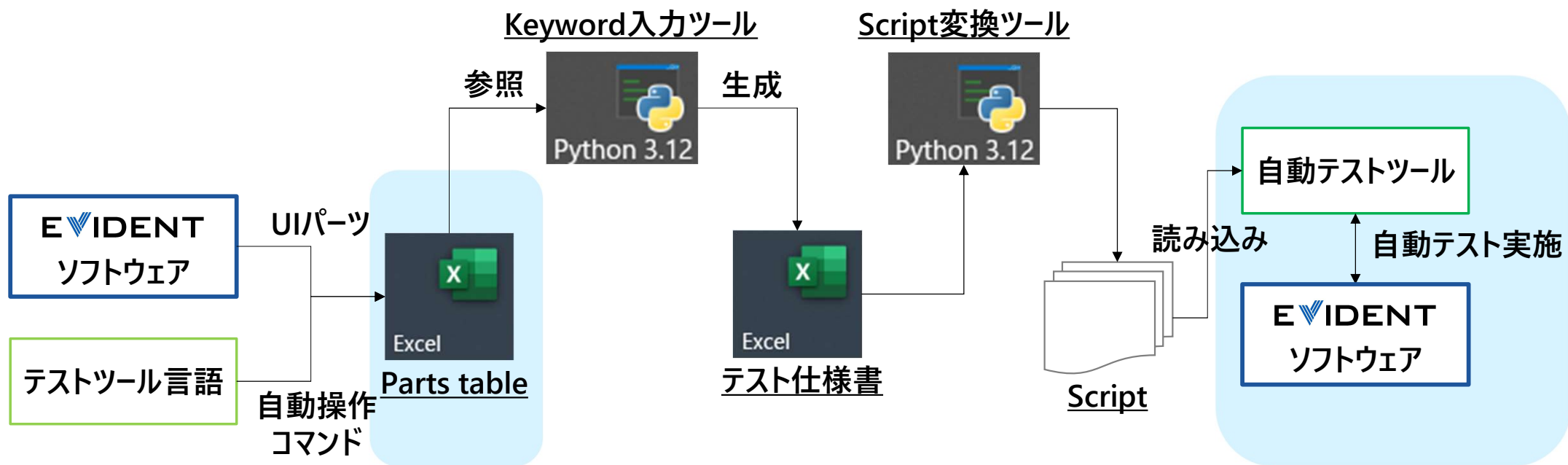


自動テストツール は市販ソフトを買い入れ

役割：自動テストエンジニア：Develop担当

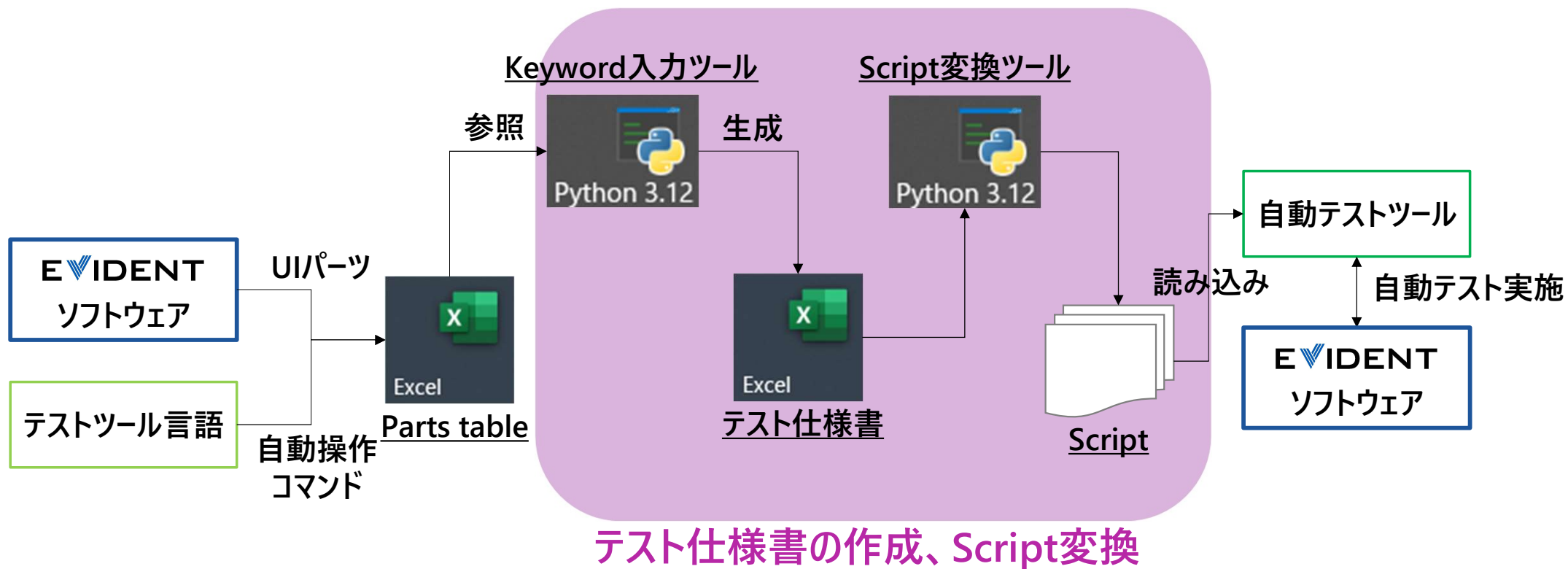


役割：自動テストエンジニア：運用担当



Parts table作成、自動テスト実行結果のデバッグ

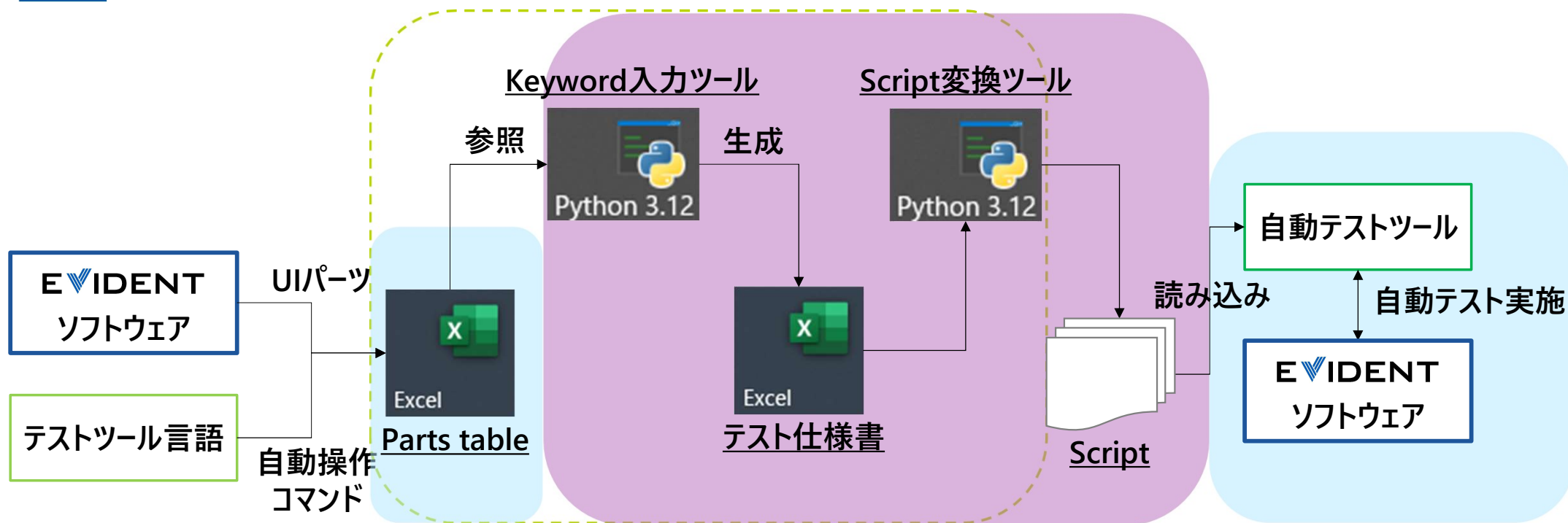
役割：テスター



Agenda

- 1 自己紹介、2022年の発表について
- 2 新たな自動テストの課題
- 3 Keyword-Driven-Testフレームワークの説明
- 4 メリット**
- 5 エ夫したこと
- 6 まとめ

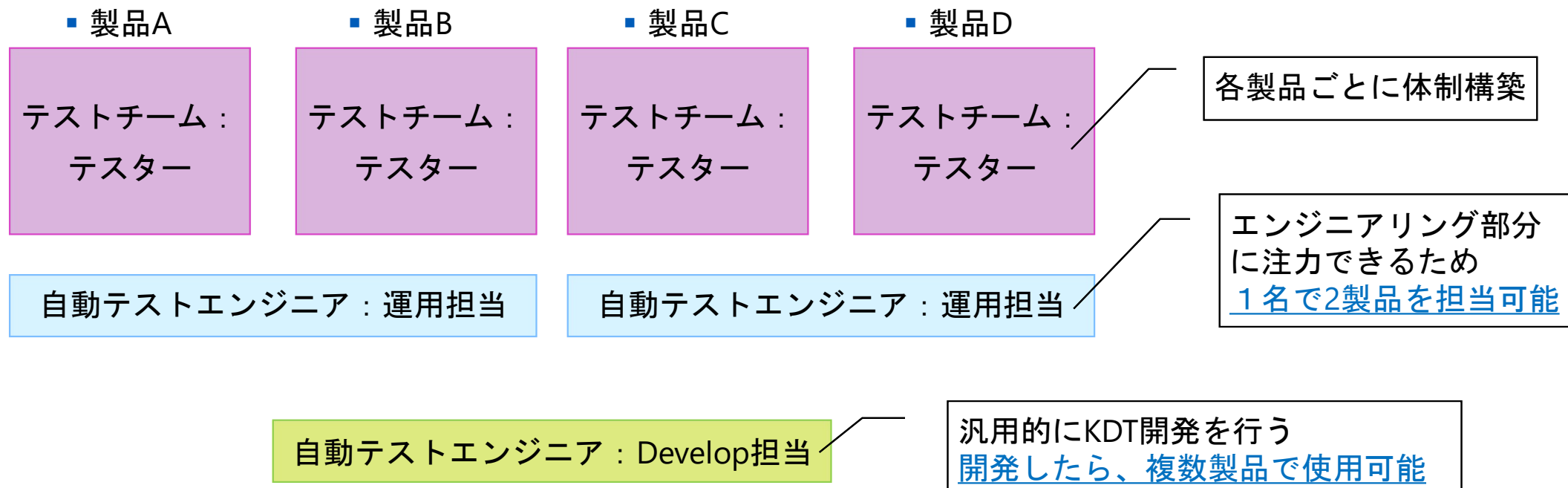
メリット 1 : 効率的な作業分担



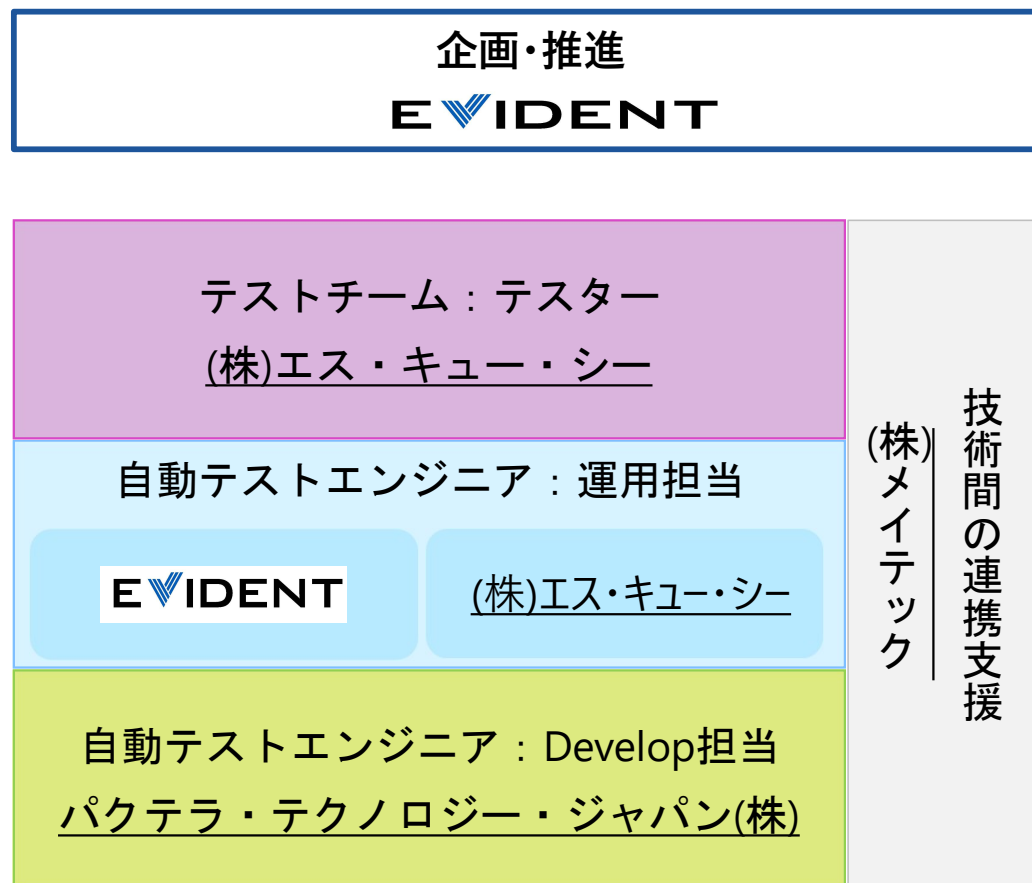
KDTにより、それぞれの得意分野に集中でき、効率的な自動化推進を可能

メリット 2 : 要員アサインの最適化

【自動テスト体制 イメージ】



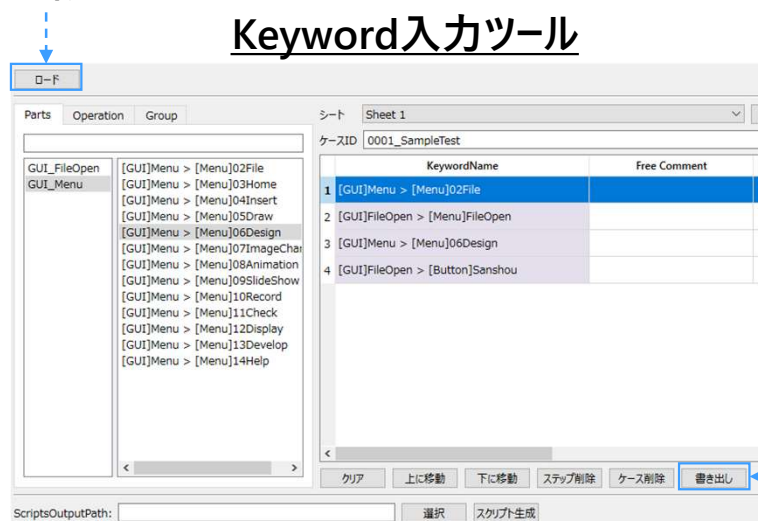
得意スキルを結集し、ワンチームでKDTを推進



メリット 3：テスト仕様書作成の生産性向上

- ・ 自作ツール及びExcelを活用することで、複数担当で並行したテスト仕様書作成が可能
※自動テストツールのライセンス制約を極小化
- ・ Keyword入力に特化したツールなので、習得コストが低い
- ・ Keyword入力ツールとテスト仕様書の互換があるため、Excel機能を使った編集が可能
検索、文字列置換、Copy & Paste

Excel読込



互換



テスト仕様書 (Excel)

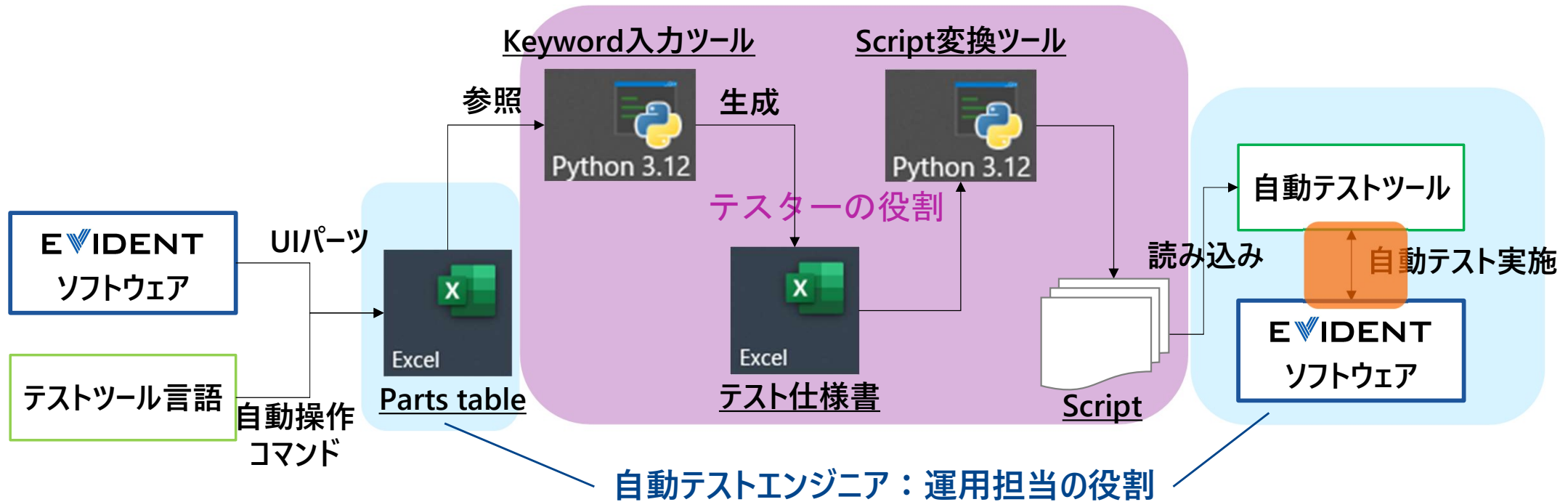
Manual Test Case ID	Manual Test Case ID		
	No.	Keyword Name	Category
0001_SampleTest	1	[GUI]Menu > [Menu]02File	Step
	2	[GUI]FileOpen > [Menu]FileOpen	Step
	3	[GUI]FileOpen > [Button]Sanshou	Step

Excel出力

Agenda

- 1 自己紹介、2022年の発表について
- 2 新たな自動テストの課題
- 3 Keyword-Driven-Testフレームワークの説明
- 4 メリット
- 5 エ夫したこと**
- 6 まとめ

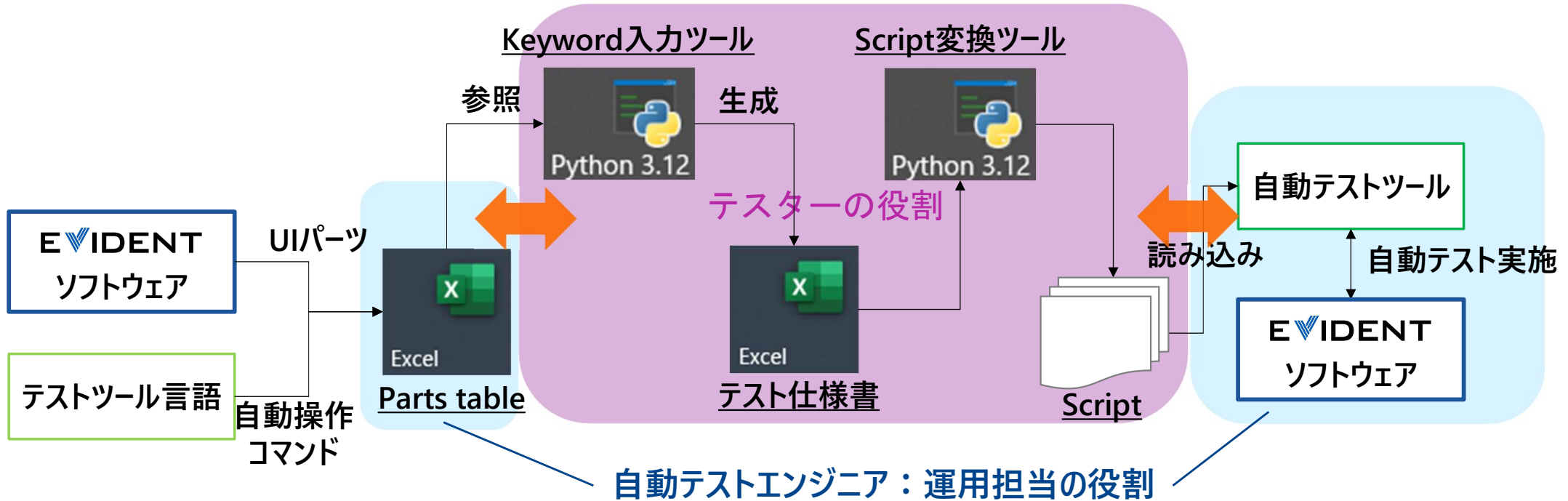
Bug切り分け



テストFail時の原因切り分けについてルール化

- 例) 1次: 自動テストエンジニアがテストスクリプトに問題ないか切り分け
 2次: テスターが同テストを手動で実施し、アプリケーションのバグか切り分け

役割の冗長性



体制が習熟したら、役割に幅を持たる

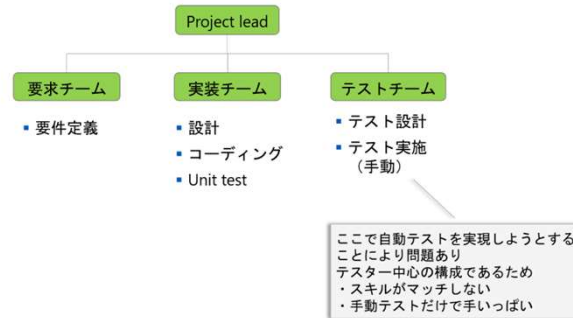
例) テスト仕様書を作成して最初の自動テストは、テストが実行失敗したときは、テスト仕様書ミスの可能性が高いため

Agenda

- 1 自己紹介、2022年の発表について
- 2 新たな自動テストの課題
- 3 Keyword-Driven-Testフレームワークの説明
- 4 メリット
- 5 エ夫したこと
- 6 まとめ

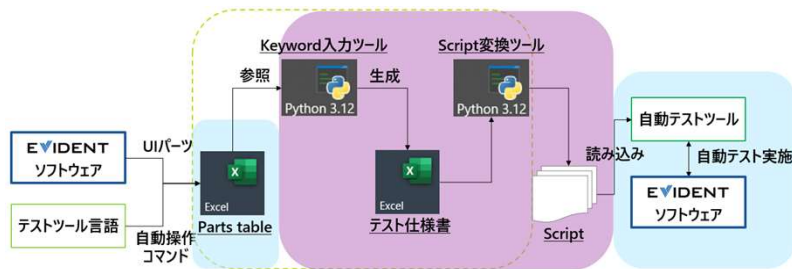
まとめ 1 : 今回の取り組み

問題：テストチームがテスター中心のため自動テスト推進が難しい



取り組み：

① KDTによる自動テストの役割分担

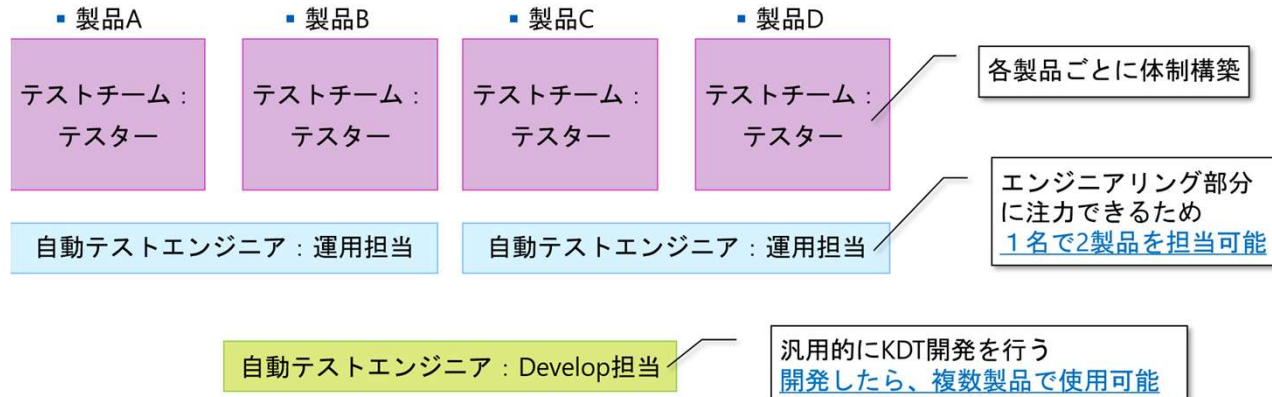


② 自動テストエンジニアの体制支援

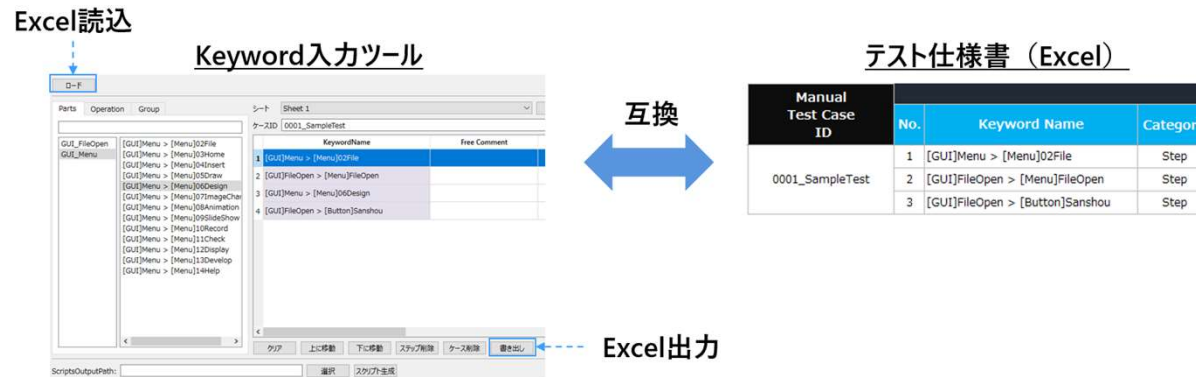


まとめ 2 : 得られたメリット

メリット：自動テストの体制を効率的に構築可能、水平展開も容易



メリット：自作ツールならではの使い勝手の良さ



おわりに：今後の目標

- ・KDTの機能拡張を行い、多くの製品へ自動テストの恩恵を深める

現在検討中：ローカライズテスト機構の開発

言語を切り替え、想定する文字列が表示されるかをチェックするテストを自動化したい

- ・KDTで得られたノウハウを整理

自動テストエンジニア、テスターの経験値は積みあがっている

自動テストエンジニア向けのノウハウは整理しつつあるが、テスター向けはまだ不十分

例) テスト仕様書の開始条件を合わせるため、イニシャライズ処理が必要

- ・KDT自動テストの経験を増やし、より多くの製品へKDTを展開したい

E  IDENT

SEEING IS SOLVING