チームで成功させるシステムテスト自動化 by Friendly

株式会社Codeer
石川達也
http://www.codeer.co.jp/

自己紹介

石川達也

株式会社Codeer代表取締役

Microsoft MVP for .Net 

Windowsアプリテスト自動化歴10年

Windowsアプリ操作用ライブラリFriendlyの開発者

http://www.codeer.co.jp/

http://ishikawa-tatsuya.hatenablog.com/

Friendly紹介

じわじわ来てます。
一部上場企業様でも続々と採用中

IPA Better Life with IT 情報処理推進機構

先進的な設計・検証技術の適用事例報告書
2014年度版に事例掲載予定

Friendly紹介

アメリカでも大好評でした！

Microsoft MVP Showcase 2位！

http://blogs.msdn.com/b/mvpawardprogram/archive/2014/11/04/mvp-showcase-winners.aspx

アジェンダ

・テスト自動化とは

・テスト自動化に必要な能力

・テスト自動化プログラムのレイヤ化

・アプリケーションドライバの実装

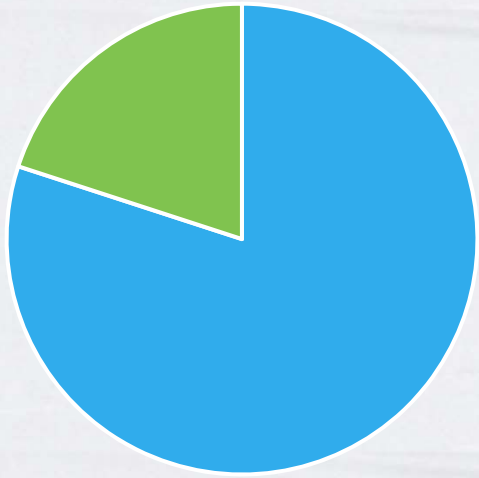
テスト自動化とは

テスト自動化

膨大なコストが必要とされる
テスト作業の

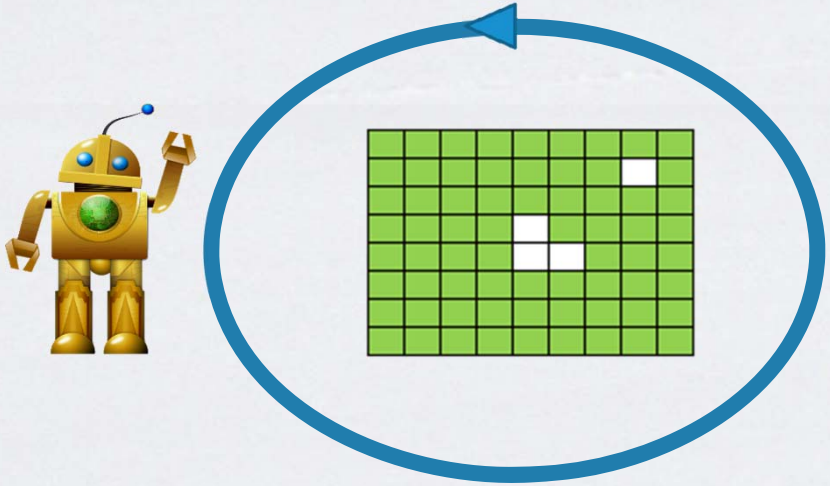
「何割か」

を自動化することです。



全てを置き換えるものではない。
非現実的な期待をしない。

自動化しておけば、テストは繰り返し実行可能です。

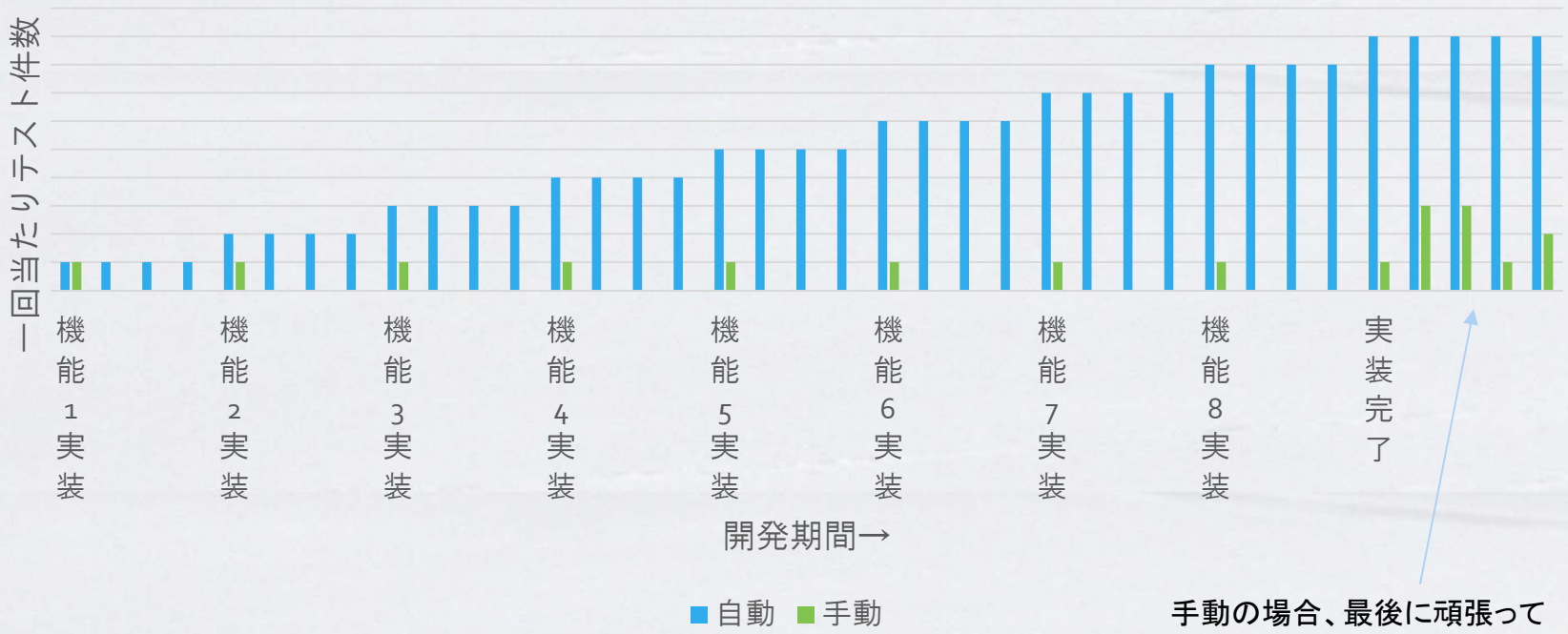


デグレードのリスク回避

【品質状況の把握】
日々、指定のテストケースに関しては、リスクが排除されていることが、分かります。
つまり、品質の状況を把握できるのです。これは自動化した最大のメリットです。

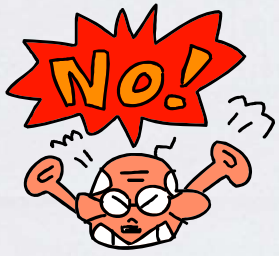
テスト自動化のメリット

テストの蓄積と、テスト実行作業の負荷分散



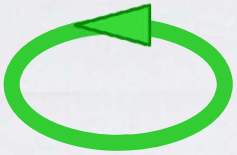
手動は、一度やったら終わりですが、
自動化しておけば、それは開発期間中ずっと実行され
デグレードのリスクを回避することができます。
テストの負荷分散にもつながります。

つい先日も、こんなことがありました。



・リリース後に重要な機能で不具合が発覚！
　その機能は網羅的に自動テストを付けていたが、
　漏れていたケースがあった。

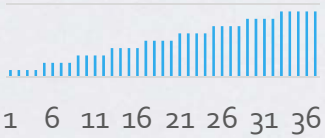
・漏れていた観点を元にテスト付け足し。



・修正後テスト実行
　→既存の自動テストでNG発見
　　→作ってて良かった！

・再修正
　→全てのケースをクリア→無事リリース！

これが可能だったのは



蓄積され続けたテストケースがあった。

そもそも網羅性の高いテストが
自動化されている機能だった。

テストケース（網羅性）が把握できていて
漏れが見つかった場合、すぐに改善出来た。

安定動作する自動テストだった。

codeer

テスト自動化に必要な能力

システムテスト自動化には、二つの能力が必要です。

テスト設計能力

テストシナリオ作成能力

テスト設計能力

・基本は手動の場合と同じ

・開発側からの視点を追加し、
グレイボックス的なメニューもあれば
より効率的

なんにせよ、
既にできているはずですよね？

テストシナリオ作成能力

ココがボトルネックになっている！

テストシナリオ作成能力

全部テストチームでやる？

・キャプチャリプレイ使う？
・スクリプトの整形は？
・トラブル発生したら？
・開発側が内部的に何か変えたときのメンテは？

無理じゃない？

テストシナリオ作成能力

全部開発チームでやる？

・調べたらできるかもしれんけど・・・
・時間ない
・量は作れない
・とは言え、やっぱやり方わからん

なしではないけど、もっといい手段ない？

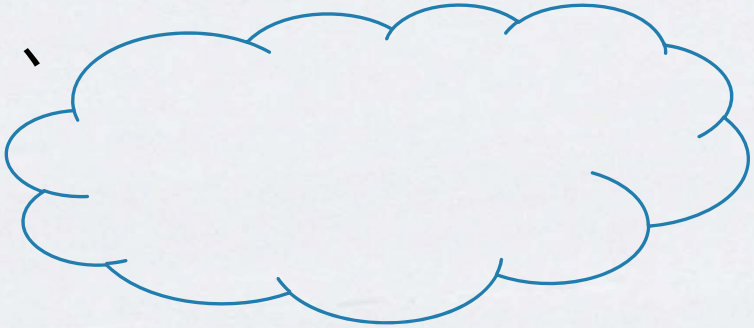
プロジェクトで一丸となって取り組めば何とかなるのでは？

テスト自動化プログラムをレイヤ化する

モヤモヤしたテストシナリオ

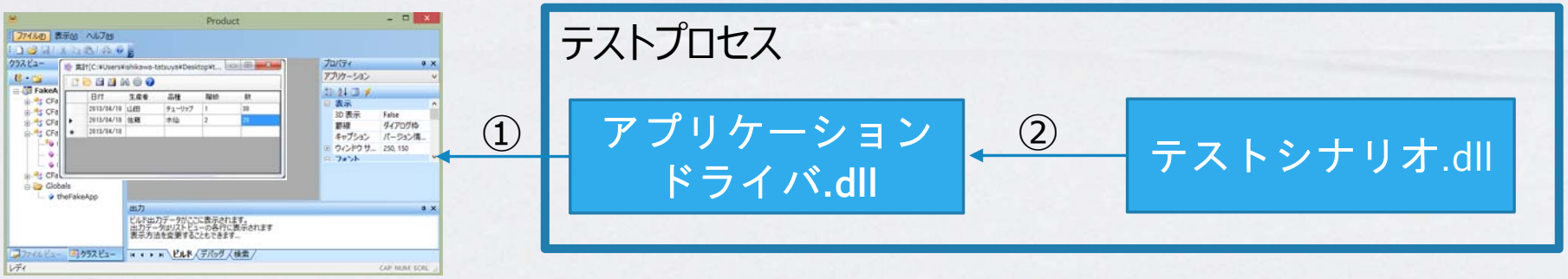
複雑な操作、
内部仕様（コントロールのIDなど）、
テスト仕様

が混沌と書かれている



メンテも量産も
もちろん分業も無理

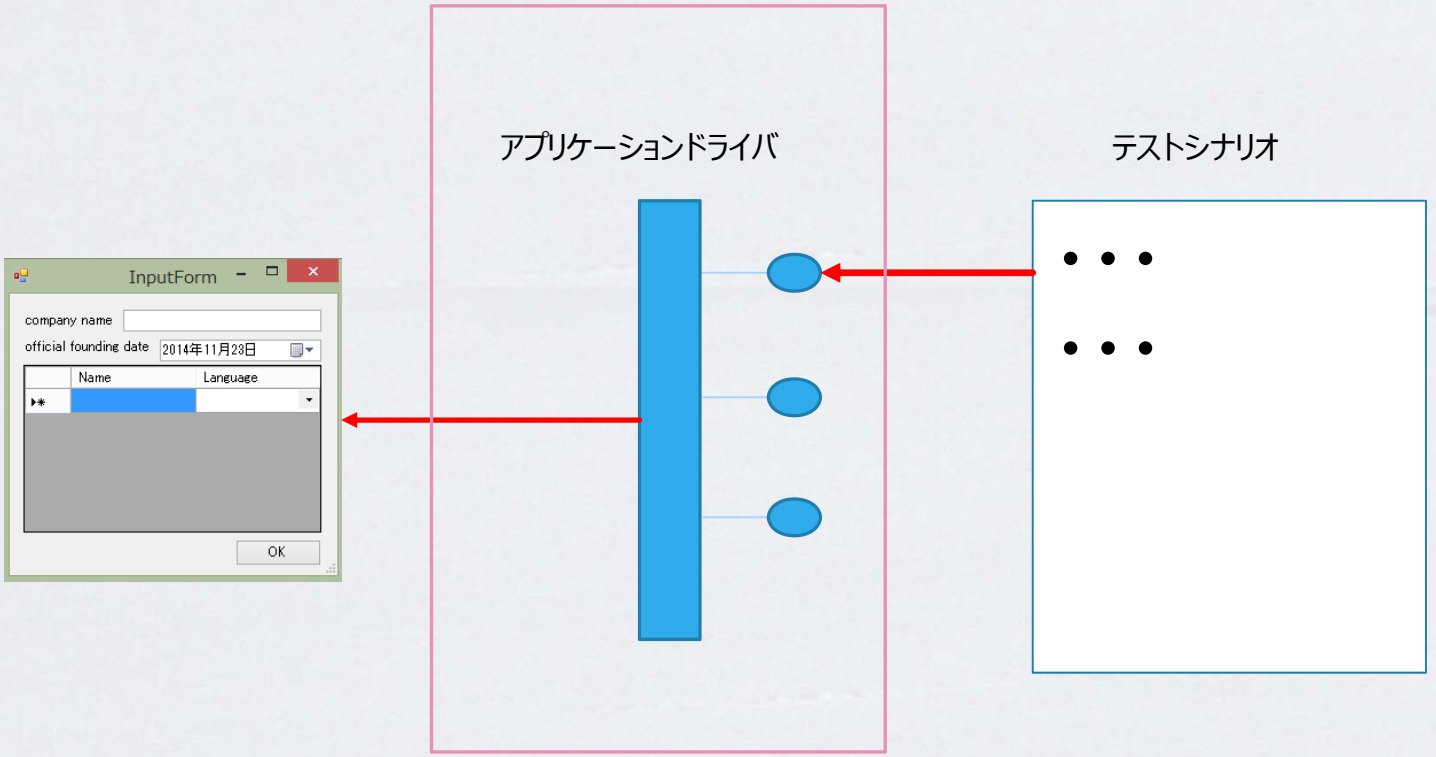
アプリケーションドライバを使ってレイヤ化したテストシナリオ



操作とテストが分離されている

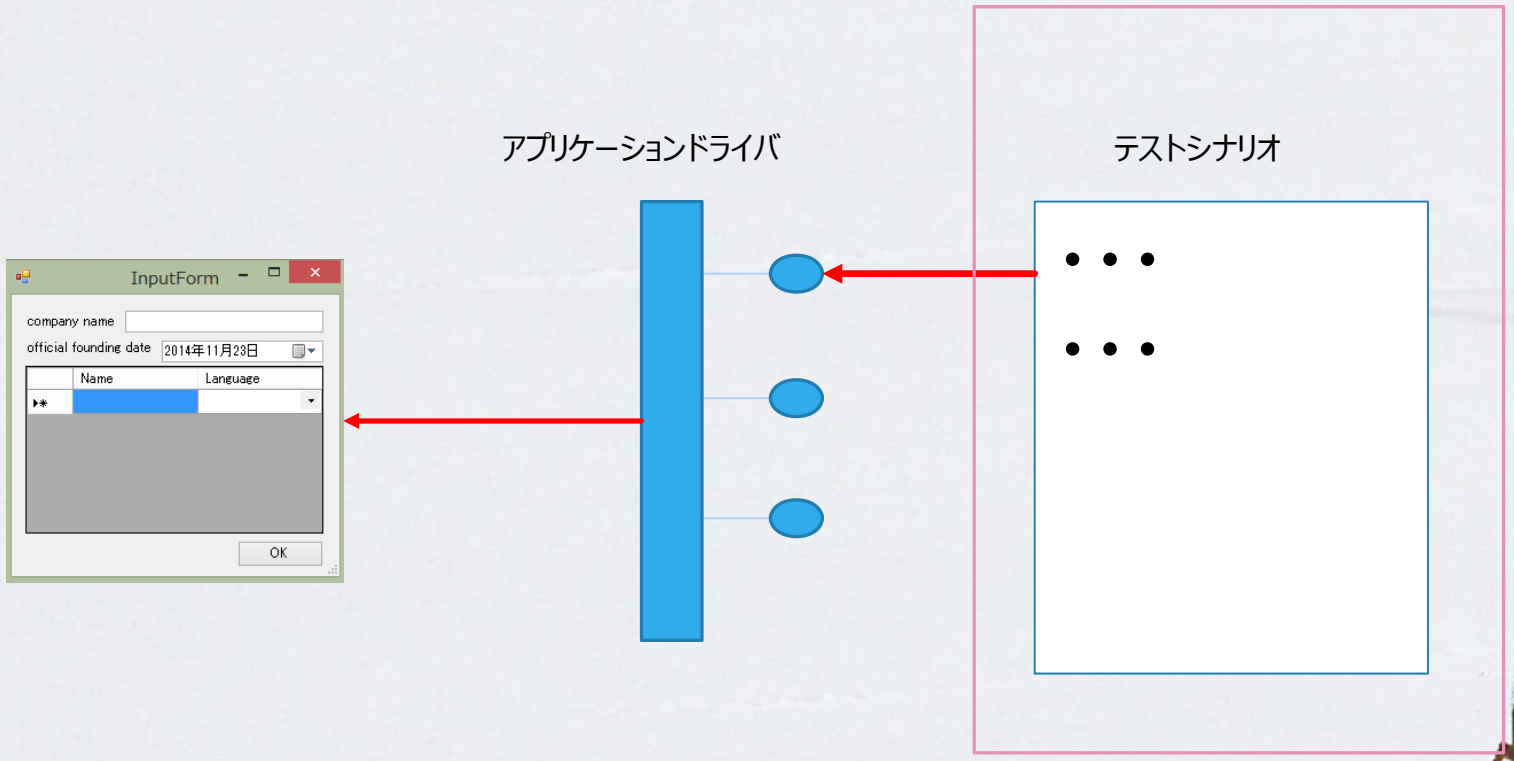
アプリケーションドライバ

アプリケーションをテストシナリオから
操作するためのインターフェイス
操作に関する複雑さを隠蔽する。
開発チームが担当。



テストシナリオ

外部仕様のみで記述される。
基礎的なプログラムの知識のみで記述可能。
テストシナリオはそれなりのボリュームになる。
主にテストチームが担当。



ボリュームイメージ

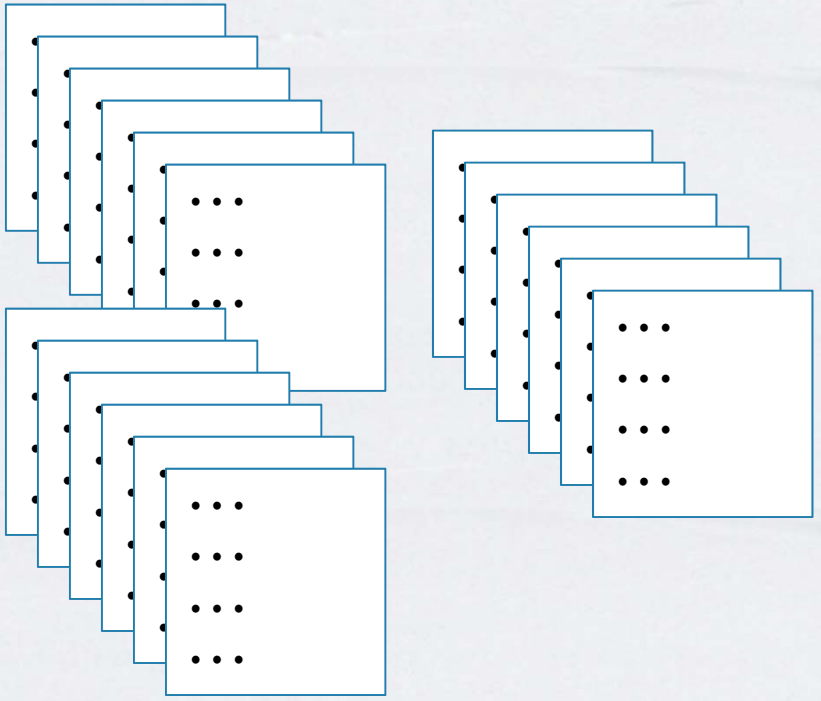
小

アプリケーションドライバ

アプリケーションドライバは
画面、機能の実装時に
付け足し実装していく

大

シナリオ



プロダクトに特化した高品質な
インターフェイスを使うことで
シナリオを容易に実装

デモ

アプリケーションドライバを使って、
テストチームがテストシナリオを書く

https://github.com/Ishikawa-Tatsuya/HandsOn14

アプリケーションドライバの実装

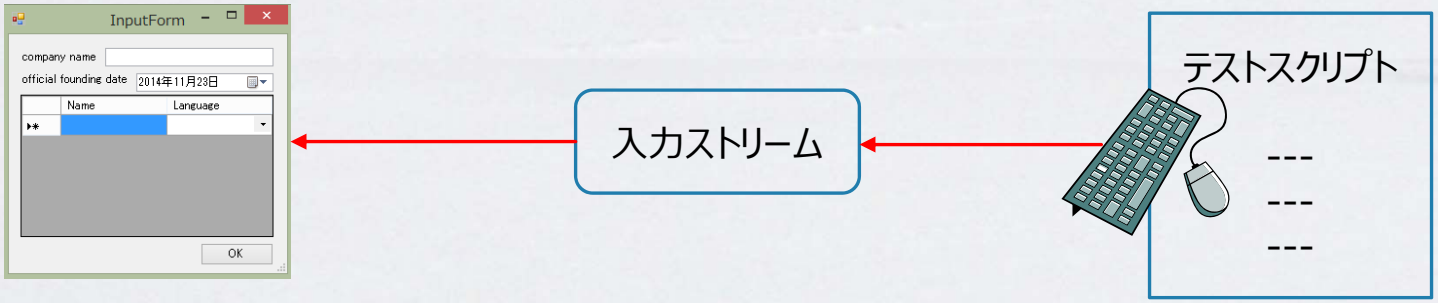
とは言え、別プロセス操作。

どうすれば？

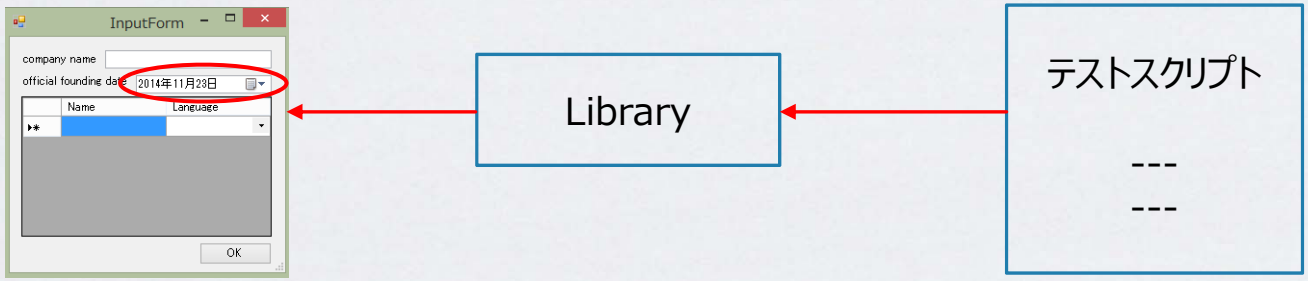
しかも低コストで作成しなければならない。

一般的には

キーマウスエミュレート
→不安定、プログラムから使うには指定が困難。

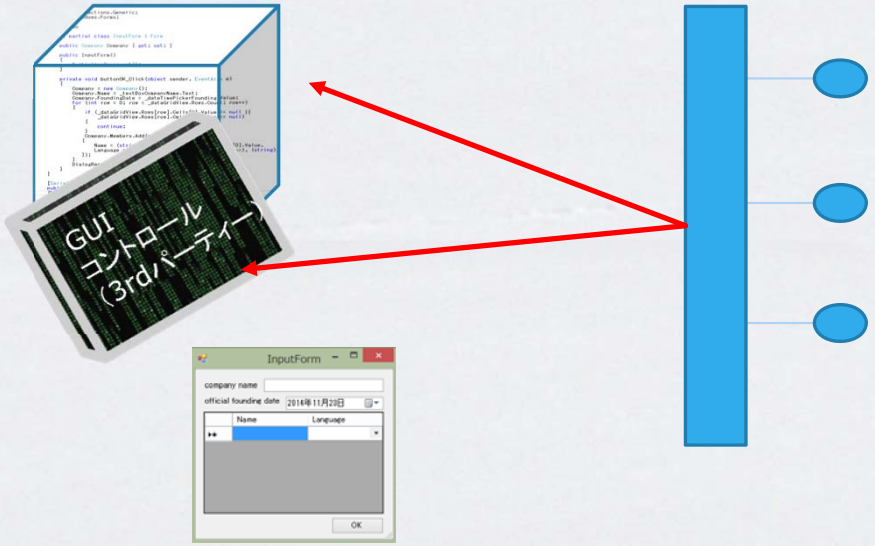


UIオートメーション
→難易度が高い。自由度が低い。



codeer

【提案】 内部APIを使いましょう！



内部APIは、ここでは通常の実装時に使用、
作成するAPIを指します。

非常に強力です。

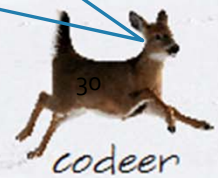
確実に動作します。

プロダクト実装時の知識を
活かせるのです！

Windowsアプリで
内部API操作はこれ！

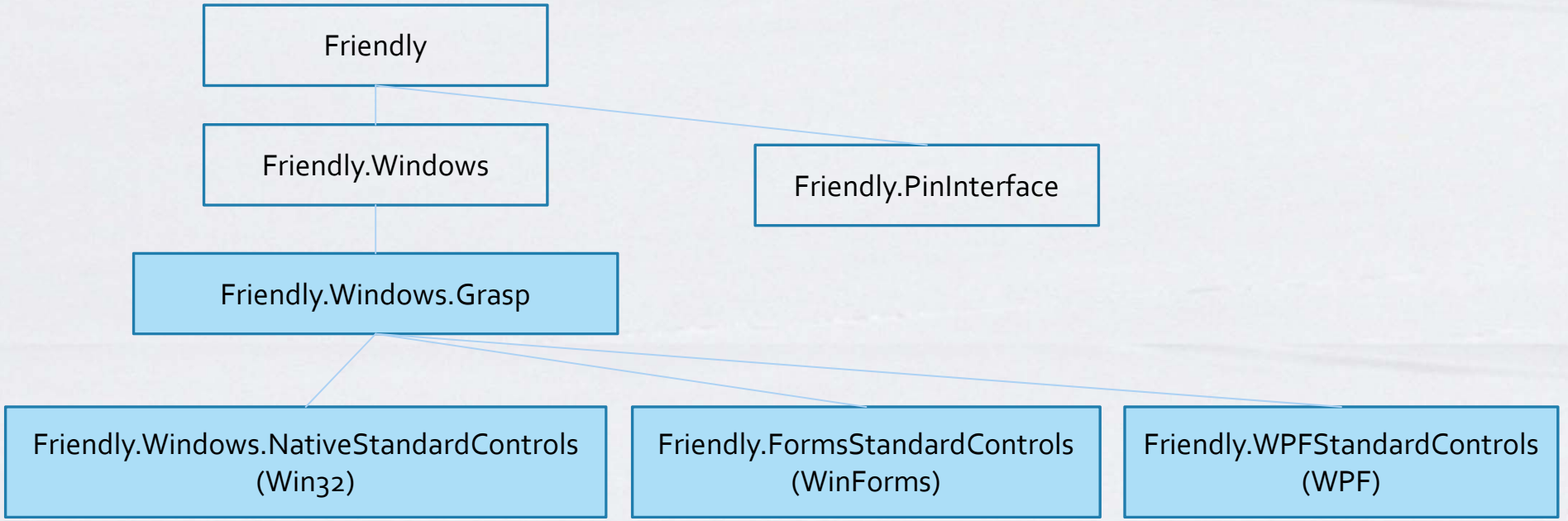
Is a magical library!
It break through
the walls of processes.

Win32、WinForms、WPF
対象プロセスに仕込みなしで
内部APIを呼べる魔法のライブラリ！



デモ

Friendlyはアプリケーションドライバを実装するのに使います。



便利なライブラリも多数提供
実装コストをさらに軽減

システムテストは団体戦です。

→あと長期戦です。

テスト自動化支援業務承らせていただきます。

・コンサルティング
・トレーニング
・テスト自動化受託
・プロダクト開発＋テスト自動化受託

ご好評いただいております。
まずは、御社の開発状況を伺わせてください。

ご清聴ありがとうございました！

【Picture】
Dawn Huczek