スプリント3: 仮想テスト

スプリント 2 の振り返りで、次のことを決定しました。

> 次のスプリントでは、インストーラーを安全にテストする方法を検討しましょう。

テストは、あらゆるアジャイル ソフトウェア開発方法論の重要な部分です...しかし、私たちはこれまでほとんど無視してきました。今こそ、ピアニストに報酬を支払い、正しいことを始めて、テスト、テスト、テストを繰り返すときです。これがスプリント 3 のテーマです。スプリントのタスクは次のとおりです。

- テストの計画
- インストーラーを実行する...勇気があれば
- Windows サンドボックスを有効にする
- サンドボックスでのテスト

このスプリントでは、スプリント終了基準がアップグレードされます。スプリント テスト プランを作成して実行する必要があります。運が良ければ、テストは成功します。

この章のページ

‹ スプリント 3: 仮想テスト　上　スプリント 3: テストの計画 ›

スプリント3: テストの計画

最後にインストーラーを終了したとき、インストーラーを少しリファクタリングして名前を整理し、プレースホルダー アプリをインストールしました。(実際のアプリとは? UI フレームワークと開発言語を絞り込むための「オフサイト計画」の後、開発チームは反対のことを行い、使用したい 3 つの UI フレームワークと 2 つのクールな新しい言語を追加しました。彼らが私たちに追いつくまではまだ時間があります。) これまでのところ、プレースホルダー アプリのインストールは、私たちが行った唯一の「実際の」機能です。本当にそうでしょうか?

WiX で Windows インストーラー パッケージを作成する利点の 1 つは、Windows インストーラーのすべての機能にアクセスできることです (WiX のすべての機能にもアクセスできますが、これについては後で詳しく説明します)。たとえば、MSI パッケージには、MSI が内部のファイルを配置するために必要なディレクトリを作成するために必要なすべてのデータが含まれています。つまり、MSI がファイルをアンインストールしてディレクトリを削除するのに必要なデータも含まれています。そのために XML を記述する必要はなく、WiX でさえ舞台裏で何もしませんでした。その動作は Windows インストーラー自体の一部です。

では、インストールとアンインストールをテストできます。他に何かありますか? ああ、そうです。パッチのアンインストール失敗のロールバックをテストするまで待ちます。しかし、今のところは、この 2 つのシナリオに固執しましょう。

テスト プランの手順には、インストールとアンインストールの方法、およびチェックするディレクトリとファイルが含まれている必要があります。スプリント 1とスプリント 2での更新から、パッケージがプレースホルダー アプリをインストールする場所がわかっています `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage\App.exe` 。

> Windows インストーラーは、スクリプト化された指示 (「これらのディレクトリを作成する」、「これらのファイルをコピーする」) ではなく、インストール後にマシンをどのように表示するか (「このディレクトリは、この別のディレクトリの下に作成する」、「このディレクトリに配置するファイルは次のとおりです」) の説明に基づいて動作するため、スクリプトの命令型アプローチと比較して、宣言型と呼ばれます。 1990 年代当時、これはインストール方法の大きな変化でした。その考え方は、数千または数百万の個別のインストーラーにすべてを完璧に実行させるよりも、1 つのエンジンにすべての面倒な作業を実行させて適切に動作させる方が簡単であるというものです。

したがって、テスト計画は次のようになります。

1. インストーラーがプレースホルダー アプリをインストールすることを確認します。
    1. .msi ファイルをダブルクリックします。
    2. インストールが完了するまで待ちます。
    3. `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage\App.exe` 存在し、先ほど作成したプレースホルダー アプリであることを確認します。
    4. 開いて `Installed apps` （別名 ARP）、 `WixTutorialPackage` リストされていることを確認します。
2. インストーラーがプレースホルダー アプリをアンインストールしてクリーンアップすることを確認します。
    1. を開いて を `Installed apps` 選択し `WixTutorialPackage` 、ボタンをクリックして `...` を選択します `Uninstall` 。
    2. アンインストールを確認し、アンインストールが完了するまで待ちます。
    3. 削除されたことを確認してください `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage\App.exe` 。
    4. ディレクトリが削除されていることを確認してください `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage` 。
    5. 再度開い `Installed apps` て確認はリストに `WixTutorialPackage` ありません。

インターン生がマーケティングのためにコーヒーを配るのに忙しくなかったら、雑用を手伝ってもらえたかもしれないのに、それで十分です。

‹ スプリント 3: 仮想テスト　上　スプリント 3: インストーラーを実行する...勇気があれば ›

スプリント 3: インストーラーを実行する...勇気があれば

Visual Studio でソリューションをビルドすると、プレースホルダー アプリの .exe とそれをインストールする .msi が取得されます。

```
Build started...
1>------ Build started: Project: App, Configuration: Debug Any CPU ------
1>  App -> X:\sprint3\App\bin\Debug\App.exe
2>------ Build started: Project: WixTutorialPackage, Configuration: Debug x86 ------
2>WixTutorialPackage -> X:\sprint3\WixTutorialPackage\bin\x86\Debug\en-US\WixTutorialPackage.msi
========== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build started at 13:37 and took 03.589 seconds ==========
```

のファイルは `X:\sprint3\WixTutorialPackage\bin\x86\Debug\en-US\WixTutorialPackage.msi` インストールする準備ができています。さあ、ダブルクリックしてください。待っています。さあ、挑戦してください。ダブルドッグに挑戦してください!

私たち専門家の言うことを信じてください。.msi のインストールは完全に安全です。これは単なるファイルのインストールであり、Windows インストーラーは1999 年の Office 2000 のリリース以来、おそらく何千兆回もこれを実行しています。インストールに問題はなく、アンインストール後はクリーンなマシン (または少なくとも以前と同じくらいクリーンなマシン) になります。

それでも、ソフトウェア開発者は、開発初期のソフトウェアのランダムビルドをインストールすることに対して当然かつ適切な注意を払います。

昔は、テスト専用のマシンが 1 台か 2 台あったので、ソフトウェアが故障しても、フォーマットしてオペレーティング システムを再インストールするだけですぐに済みました。その後、仮想マシンが登場しました。これは、宇宙の慈悲深さの証拠です。仮想マシンを使用すると、デスクに座ったまま、"ホスト" マシン上で偽のマシン全体を実行できます。起動して大混乱を引き起こし、ボタンをクリックするだけでマシンが消えます。ソフトウェアの神がセットアップ開発者に、どうすれば仕事が耐えられるようになるかを尋ね、その答えをクッション付きの銀の皿に載せて渡したようなものです。

私たちはファンです。

WindowsはWindows 8以降、WindowsのクライアントエディションにHyper-Vハイパーバイザーを搭載しています。Windows 10ではWindows Sandboxが導入されました。

> アプリケーションを分離して安全に実行するための軽量デスクトップ環境を提供します。

まさに私たちが探していたものです。サンドボックスは Windows に付属しており、一般的な Hyper-V 仮想マシンよりも軽量で、起動も速いという便利な機能もあります。サンドボックスが完全な Hyper-V VM と比べて唯一不利な点は、サンドボックスではホスト マシンと同じバージョンの Windows しか実行できないことです。異なるバージョンをテストするには、おそらく仮想マシンである他のマシンが必要になります。これはいずれ問題になりますが、今のところはサンドボックスの利点が欠点をはるかに上回っているため、Windows サンドボックス内でテストを開始します。

‹ スプリント 3: テストの計画　上　スプリント 3: Windows サンドボックスの有効化 ›

スプリント 3: Windows サンドボックスの有効化

Windows Sandbox は Windows に付属していますが、デフォルトではインストールされません。ドキュメントには、機能リストを表示するにはタスクバーから を検索するように記載されています `Turn Windows Features on or off` 。そのリストにアクセスするもう 1 つの方法は、古くからある ARP (プログラムと機能) にアクセスして、 `Turn Windows Features on or off` 左側のリンクをクリックすることです。

すると、機能の一覧が表示されます。一覧のほぼ下までスクロールすると、 が表示されます。 が表示されない `Windows Sandbox` 場合は、お使いのマシンが Windows の最新 CPU への欲求を満たすほど最新ではないことを意味します。 が表示されていてチェックされている場合は、これで完了です。 が表示されていてチェックされていない場合は、チェックして [OK] をクリックします。サンドボックスがインストールされるので、再起動とスピナーの回転が必要になります。

インストールが完了したら、ショートカットを検索し `sandbox` て選択します `Windows Sandbox` 。初めて実行すると、表示されるまでに少し時間がかかる場合があります (初心者の緊張のため) が、すぐに、見慣れた Windows デスクトップのウィンドウが表示されます。

それではインストーラーを入手してテストしてみましょう。

‹ スプリント 3: インストーラーを実行する...勇気があれば　上　スプリント 3: サンドボックスでのテスト ›

スプリント 3: サンドボックスでのテスト

Windows Sandbox のドキュメントでは、ホスト マシンのファイル エクスプローラーからファイルをコピーして、Sandbox マシンに貼り付ける方法について説明しています。これはかなり手動的な作業であり、間違いなく改善の余地がありますが、開始するには最適な場所です。

したがって、ビルドした .msi パッケージを取得する必要があります。

```
Build started...
1>------ Build started: Project: App, Configuration: Debug Any CPU ------
1>  App -> X:\sprint3\App\bin\Debug\App.exe
2>------ Build started: Project: WixTutorialPackage, Configuration: Debug x86 ------
2>WixTutorialPackage -> X:\sprint3\WixTutorialPackage\bin\x86\Debug\en-US\WixTutorialPackage.msi
========== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build started at 13:37 and took 03.589 seconds ==========
```

ファイル エクスプローラーで `X:\sprint3\WixTutorialPackage\bin\x86\Debug\en-US\` ディレクトリを開きます。より正確には、パッケージをビルドしたマシン上のディレクトリです。

次の 3 つのファイルが表示されます。

- キャブ1.キャブ
- Wixチュートリアルパッケージ.msi
- Wixチュートリアルパッケージ.wixpdb

最初の 2 つが .msi パッケージを構成します。.msi ファイル自体と、(現時点では) プレースホルダー アプリのみを含む .cab キャビネット ファイルです。キャビネットを .msi ファイル自体に埋め込むこともできますが、WiX のデフォルト設定では、.msi ファイルの外部に保持されます。外部キャビネットは、特にパッケージが数十/数百メガバイトにまで大きくなると非常に便利です...残念ながら、キャビネットはわずか 4... キロバイトです。では、このコピー アンド ペーストを少し簡単にして、キャビネットを埋め込む変更をこっそりと加えましょう。

WiX では、要素はキャビネットの生成と埋め込み方法を制御します。現在、Package.wxs には要素 `MediaTemplate` がありませんが、何が起こっているのでしょうか。WiX は、常に役立つツールセットであり、要素を作成していないことを考慮して、すべてのデフォルト値を持つ要素を追加しました。要素を追加すると、WiX はその要素を使用するため、デフォルト値を簡単に上書きできます。属性のデフォルトをから に上書きして、WiX がキャビネットを埋め込むようにします。次のように、要素を要素の子として追加します。 `MediaTemplate MediaTemplate MediaTemplate EmbedCab no yes MediaTemplate Package`

```
<Wix xmlns="http://wixtoolset.org/schemas/v4/wxs">
  <Package
    Name="WixTutorialPackage"
    Manufacturer="Edgerock Concepts"
    Version="1.0.0.0"
    UpgradeCode="64deef2a-cf99-4a0c-be41-5faa802a9502">

    <MajorUpgrade DowngradeErrorMessage="!(loc.DowngradeError)" />

    <MediaTemplate EmbedCab="yes" />

    <Feature Id="Main">
      <ComponentGroupRef Id="AppComponents" />
    </Feature>
  </Package>
</Wix>
```

ソリューションを再構築すると、エクスプローラーに表示されるファイルのリストが 33% 削減されます。

- Wixチュートリアルパッケージ.msi
- Wixチュートリアルパッケージ.wixpdb

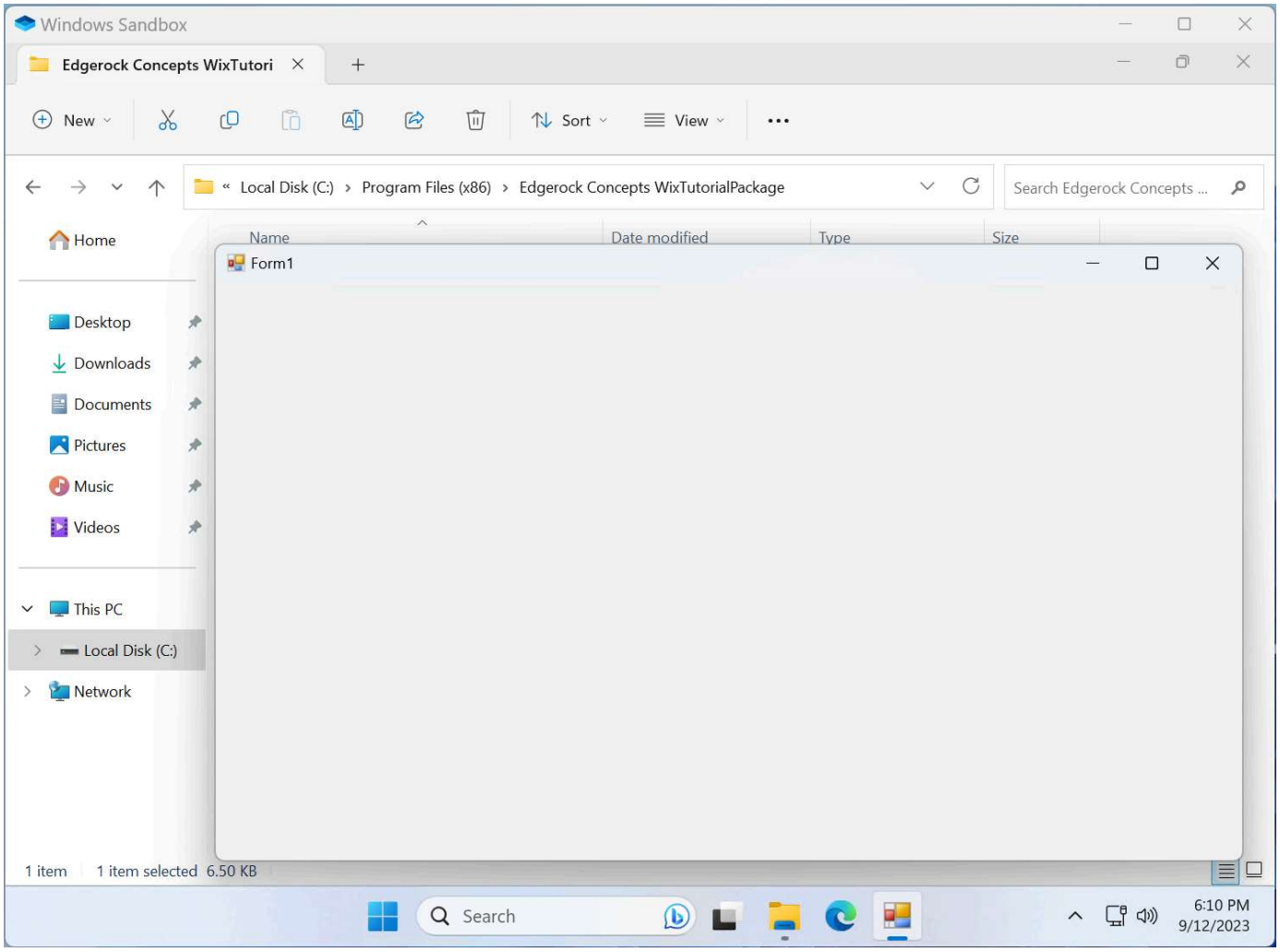
> WixTutorialPackage.wixpdb は、他のプロジェクトで見たことがあるかもしれない .pdb ファイルと同様に、デバッグ ファイルです。これには、一部の高度なシナリオで役立つ多くの追加データが含まれていますが、実際の .msi パッケージ出力の一部ではありません。今のところ、これを無視しても問題ありません。

WixTutorialPackage.msi を選択して を押します `Ctrl+C` (またはコンテキスト メニューから [コピー] を選択します)。実行中の Windows Sandbox に移動し、 `Ctrl+V` デスクトップで を押します (または [貼り付け] を選択します)。

これでテスト計画の実行を開始できます。

> 1. インストーラーがプレースホルダー アプリをインストールすることを確認します。
>   1. .msi ファイルをダブルクリックします。
>   2. インストールが完了するまで待ちます。
>   3. `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage\App.exe` 存在し、先ほど作成したプレースホルダー アプリであることを確認します。
>   4. 開いて `Installed apps` （別名 ARP）、 `WixTutorialPackage` リストされていることを確認します。

.msi をダブルクリックすると、進行状況バーが表示されたウィンドウがほんの一瞬表示されます。これは当然のことです。今のところインストールするものはほとんどありません。ファイル エクスプローラーでは、確かにディレクトリがあり `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage` 、その中に があることがわかります `App.exe` 。これをダブルクリックすると、空白のウィンドウが表示されますが、これは間違いなくプレースホルダー アプリです。



検索すると、インストールされたアプリとしてその名誉ある位置を占めるウィンドウが `apps` 表示されます。

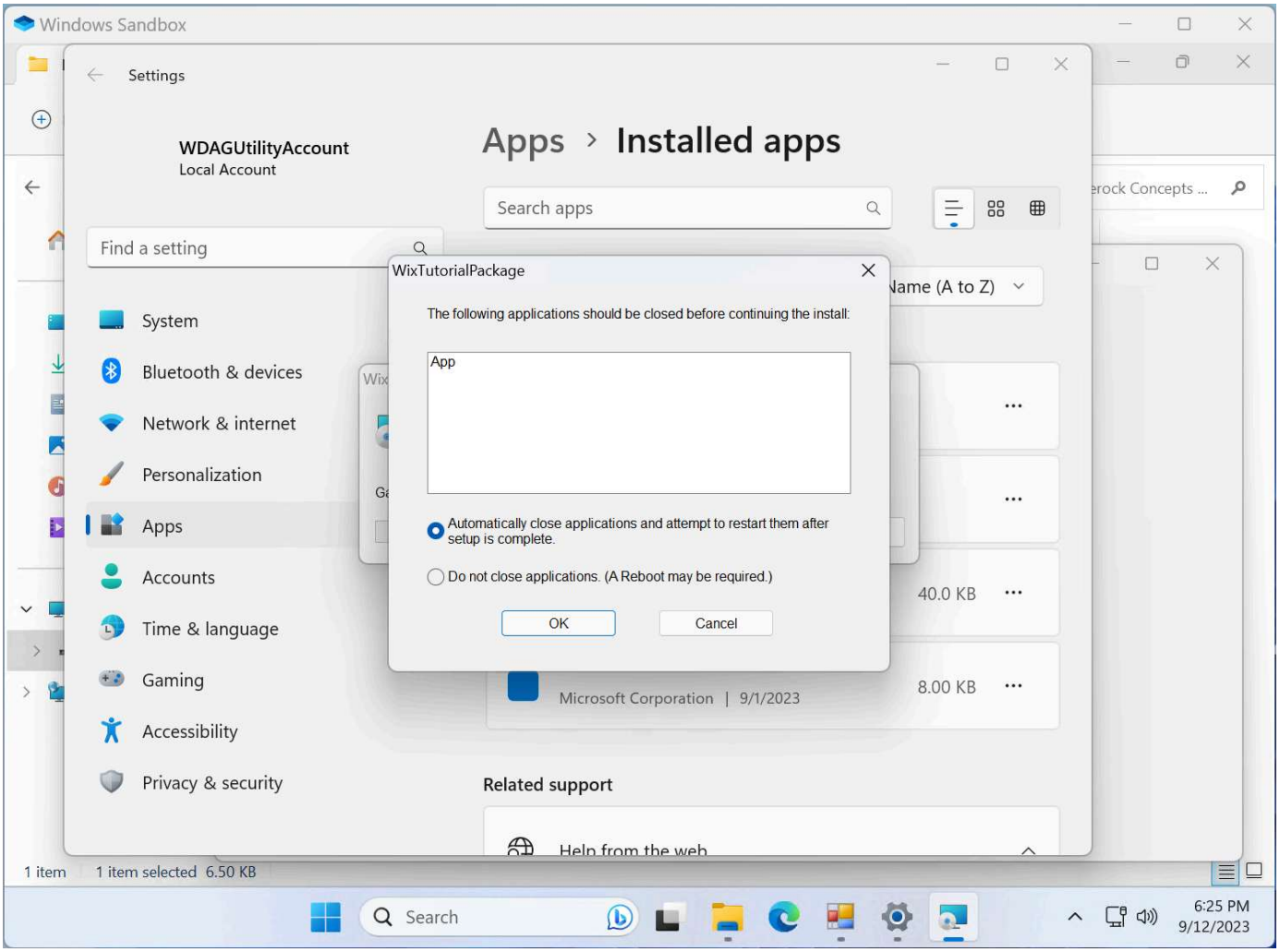
```
Installed apps WixTutorialPackage
```

ここまでは順調です。テスト計画の後半では、私たちが行ったことを元に戻します。

> 1. インストーラーがプレースホルダー アプリをアンインストールしてクリーンアップすることを確認します。
>   1. を開いて を `Installed apps` 選択し `WixTutorialPackage` 、ボタンをクリックして `...` を選択します `Uninstall` 。
>   2. アンインストールを確認し、アンインストールが完了するまで待ちます。
>   3. 削除されたことを確認してください `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage\App.exe` 。
>   4. ディレクトリが削除されていることを確認してください `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage` 。
>   5. 再度開い `Installed apps` て確認はリストに `WixTutorialPackage` ありません。

すでに開いている `Installed apps` ので、アンインストールを開始しましょう。ほぼ確実に成功します。

まあ、そうでもないですね。



テスト プランにギャップがあることが分かりました。プレースホルダー アプリを実行したままにして、アンインストールしようとしているアプリを閉じるように求める Windows インストーラーのプロンプトを考慮していませんでした。

「OK」をクリックして、デフォルト設定が正しいことを祈りましょう。アンインストールが完了し、アプリ リストのエントリが消えるので、問題ありません。ファイル エクスプローラーでざっと確認すると、ディレクトリが削除されている `WixTutorialPackage` ことがわかります。アンインストールは正常に完了しました。 `C:\Program Files (x86)\Edgerock Concepts WixTutorialPackage`

スプリント 3: Windows サンドボックスの有効化 上 スプリント3の振り返り ›

スプリント3の振り返り

スプリントの振り返りはスプリントを終了し、チームが何がうまくいったか、何がうまくいかなかったか、そしてチームがどのように改善できるかについて正直に話し合う機会を与えます。スプリント 3 では、どうでしたか?

何がうまくいきましたか?

インストーラーを実行し、Windows Sandbox を使用して安全に実行しました。

もっとうまくいく方法は何でしょうか?

Windows Sandbox の使用は、石のナイフや熊の皮のように、手で切り取って貼り付けるという、かなり手作業です。今のところは問題ありませんが、その面倒な作業の一部を自動化できれば便利です。その分野ではいくつかのオプションがあるので、テストするより本格的なインストーラーができたら、今後のスプリントで検討する予定です。

次のスプリントで何を改善しますか?

インストーラー プロジェクトにちょっとした調整をこっそり加えましたが、そろそろ WiX オーサリングを書き始める時期かもしれません。賛成の方は「賛成」と言いましょう。「賛成」の人が勝ちです。

‹ スプリント 3: サンドボックスでのテスト　上　スプリント 4: ブロッキング パッケージ ›