

デジタル トランスフォーメーション 実現のためのプレイブック

ローコードデジタルファクトリーによる
デジタルジャーニーの推進



目次



はじめに 4

本ガイドの対象読者	4
本ガイドの目的.....	4
デジタルトランスフォーメーションのユースケース	5
変革のためのフレームワーク	6

デジタルジャーニーのステージ 7

基礎	7
センターオブエクセレンス.....	7
フルデジタルエンタープライズ.....	7

基礎固め：ローコードデジタルファクトリーのセットアップ 8

構造	9
人材	14
エコシステム	16
プロセス.....	20

まとめ 24

付録 26

コンピテンシーのスキルレベルの内訳.....	26
コンピテンシーの内訳.....	27
作業項目の定義.....	30



はじめに

組織が成功を収めるためには、繰り返しの作業で溢れる環境から、価値が常に流動化する世界に対応できるようにしなければなりません。これは、人間の共同作業に関する農業革命以来の最大の構造転換です。”

—ビル・ドレイトン（社会起業家）



本ガイドの 対象読者

ローコードがもたらすビジネスチャンスに関心のある方は（中には、すでにローコードの取り組みを始めている方もいるでしょう）、このガイドを参照いただくと、この取り組みをどのように進めていくかをご確認できます。本ガイドは、2001年以降にOutSystemsが課題解決を支援してきたお客様の経験と、OutSystemsがお客様と共に開発したソリューションに基づいています。

ほぼすべての業界にわたって長年をかけて実証されてきた方法論を活用することができるため、本ガイドは大きな助けとなるはずです。

本ガイドの目的

本ガイドでは、ローコードデジタルファクトリーと呼ばれる新しい方法論を紹介します。これは、組織がデジタルトランスフォーメーションイニシアチブのあらゆる段階に適用できる方法論です。デジタルトランスフォーメーションイニシアチブでは、そのユースケース、変革のためのフレームワーク、およびデジタルジャーニーのステージという3つの概念が核となります。これらの概念に対応するために使用できるステップを詳細に説明します。

デジタルトランスフォーメーションのユースケース

デジタルトランスフォーメーションを容易にするために利用できるローコードプラットフォームには、さまざまなものがあります。そのために、ほとんどのデジタルトランスフォーメーションの取り組みが複雑になっています。多くの場合、デジタルトランスフォーメーションは、パイモダルの概念により解釈されるか、または特定のプロジェクト（データの移行など）に限定されたものになっています。しかし、デジタルトランスフォーメーションには、はるかに大きな機会があります。実際、デジタルトランスフォーメーションには、デジタルオペレーション、デジタルエクスペリエンス、デジタルコアという3つの明確なユースケースがあります。

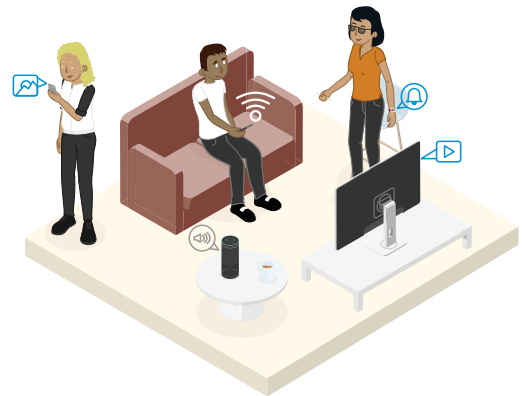
デジタルオペレーション

デジタルトランスフォーメーションは社内から開始できます。つまり、社内業務をサポートするアプリケーションを対象とします。無理矢理まとめ上げられ、手作業、電子メール、スプレッドシートなどに依存しているものを置き換えることが目標になります。典型的なソリューションとしては、ダッシュボード、ワークフロー、Webポータル、モバイルアプリケーション、中小規模のデータベースアプリケーションを挙げることができます。デジタルオペレーションのニーズには、レスポンスなモバイル/Webアプリケーションの迅速な開発、フォームやワークフローの単純な開発、記録システムとの迅速な統合、迅速な変更と展開、管理者が生産性を把握できるようにするためのインサイト提供の仕組みやダッシュボードなどがあります。



デジタルエクスペリエンス

もう1つのユースケースは、顧客のデジタルエクスペリエンスの改善です。典型的なソリューションは、顧客のモバイルアプリケーション、顧客ポータル、およびビジネスポータルです。現場作業用アプリケーションやスポーツイベント用アプリケーションなども含まれます。デジタルエクスペリエンスを構築するには、高速のプロトタイピングと開発、ピクセルパーフェクトなデザイン、大規模なスケーラビリティ、およびオフラインデータを含むセキュリティが必要です。さらに、これらのエクスペリエンスを計測して、継続的な改善のためのインサイトを取得します。



デジタルコア

デジタルトランスフォーメーションの最終的なユースケースは、古いシステムと高度にカスタマイズされたパッケージの組み合わせに対応することです。これらのコアシステムは、ビジネスニーズに対応できない場合があります。また、置き換えるには何年もかかる可能性があります。このような状況では、典型的と言えるソリューションはありませんが、当社のお客様の事例として、運送/流通業界のターミナル管理システム、消費者製品業界のERPシステム、ライフサイエンス分野の臨床試験管理システムの構築などが該当します。

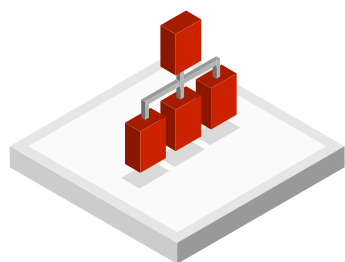
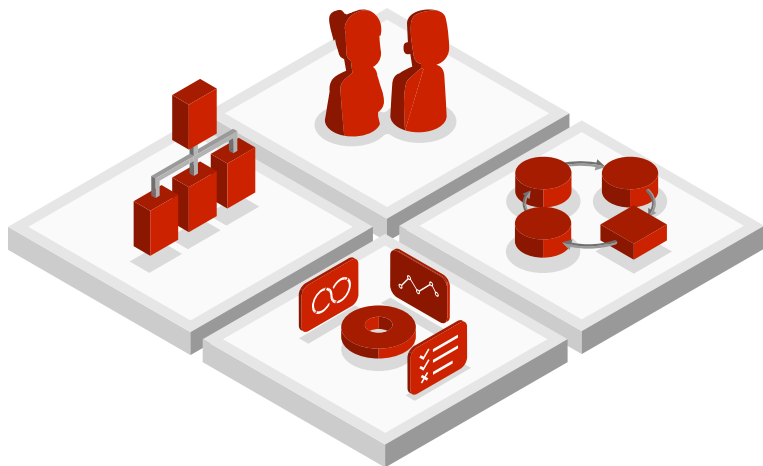


概念1：

ローコードプラットフォームで達成できる成果の範囲は、大きく分けて3つのユースケース（デジタルオペレーション、デジタルエクスペリエンス、デジタルコア）として示されます。

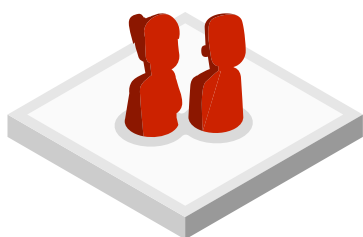
変革のためのフレームワーク

どの組織にとっても、デジタルトランスフォーメーションは大規模な取り組みとなり、「これまでのビジネス手法」を大きく変えることとなります。ローコードデジタルファクトリーと呼ばれるフレームワークを使用することで、このような劇的な変化に効果的に対処できます。このフレームワークにより、イニシアチブを推進するための構造を確立しながら、俊敏に変化に対応できるようになります。このフレームワークは、以下に示す構造、人材、エコシステム、プロセス（STEP）という4つの次元で構成されます。



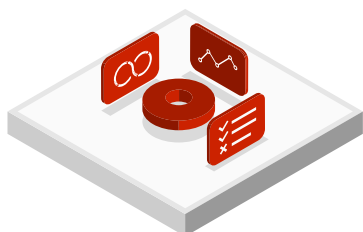
S (Structure : 構造)

デジタルファクトリーが提供する組織構造は、ローコードの開発とデリバリーを成功させるために特別に設計されたものです。チームの規模、チームの主な役割、アーキテクチャーを適用する場所、ガバナンス、テストの問題に対応します。



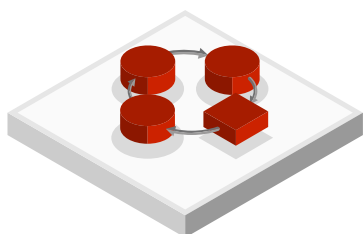
T (Talent : 人材)

ローコードの導入で期待されるのは、専門性の高いリソースを必要とせずに、消費者向けのモバイルエクスペリエンスを含め、洗練されたソリューションを構築することです。デジタルファクトリーにおける人材の次元は、ローコードプロジェクトのデリバリーを成功させるために役割を特定/調達する慎重なアプローチになります。



E (Ecosystem : エコシステム)

ローコードプラットフォームは、単独で存在するものではなく、記録、データサービス、サービスバス、クラウドサービス、ライフサイクル、DevOpsツールなどのシステムと統合されます。エコシステムの次元は、統合に取り組む順序を検討するときのガイダンスとなります。



P (Process : プロセス)

変更管理フレームワークを完成させるためには、ソフトウェアの新しいデリバリー方法をサポートするために変更/確立する必要のあるプロセスを考慮する必要があります。デジタルファクトリーにおけるプロセスの次元は、組織のさまざまな要素による相互作用のあり方、ローコードをアジャイル開発に適用する部分、成功のためにビジネスとITが連携する方法を規定します。

概念2 :

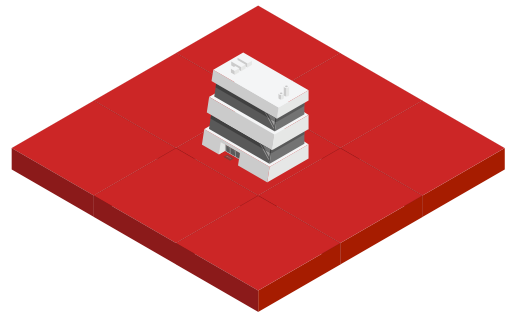
ローコードデジタルファクトリーは、ローコードテクノロジーの導入の成功を促進します。ローコードのメリットを活用するために、組織構造の提案、適切な人材の特定と採用、幅広いテクノロジーエコシステムへのローコードの適合、プロセスの最適化を行います。

デジタルジャーニーのステージ

デジタルトランスフォーメーションには3つのステージがあります。ローコードのデジタルファクトリーは各ステージに対応しているので、次のステージへの移行が容易です。

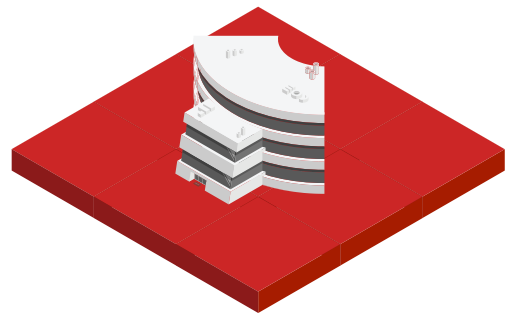
基礎

このステージでは、最初のプロジェクトチームを立ち上げ、最初のプロジェクト（デジタルオペレーションまたはデジタルエクスペリエンスのユースケース）でデリバリーを成功させることに焦点を当てています。ローコードを自信を持って活用できるようになり、さらなるプロジェクトに取り組むことができるように、早い段階で成功を収めることが重要です。



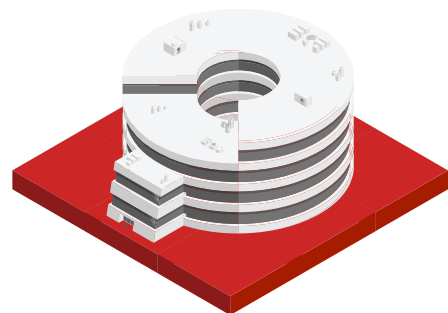
センターオブエクセレンス

最初のチームがプロジェクトの成功を積み重ねていくのに伴い、自然にローコードの活用が増大したり拡張したりすることになります。ローコード開発のセンターオブエクセレンス（COE）を構築することで、さらに多くのチームがローコードを活用できるようになります。また、ローコードを他の社内プラットフォームと容易に統合できるようになり、新たにガバナンスを重点項目として追加できます。このステージで、組織は自立的なローコード開発能力を確立します。



フルデジタルエンタープライズ

組織は、適切なサポート体制を確立することで、複数の事業部門の間でローコードを移行して成長させることができます。フルデジタルエンタープライズを目指す取り組みにより、ローコードのすべてのユースケースに対応できるようになります。極めて俊敏なデジタルコアを確立することで、ビジネスは変化する市場の状況と競争圧力にも迅速に適応できます。



概念3：

組織におけるローコードの使用は、3つのステージを経て成熟します。基礎レベルでは、最初のチームを立ち上げ、最初のプロジェクトを成功させることに重点が置かれます。センターオブエクセレンスのステージでは、複数のチームでローコードを導入し、組織は自立的なローコード開発能力を構築します。フルデジタルエンタープライズは、複数の事業部門にわたってローコード能力を拡張します。

基礎固め：ローコード デジタルファクトリーの セットアップ

新しい方法論を確実に導入するための最善の方法は、実際に効果があると示すことです。

デジタルトランスフォーメーションの基礎ステージでは、多くの人の支持を獲得し、批判を抑えます。業務を円滑にしたり、消費者に好印象を与えたりする実用的なアプリケーションを作り出します。しかも、それを迅速に実現する必要があります。ローコードデジタルファクトリーにより、この基礎固めを達成し、最初のアプリケーションを迅速に作成できます。

基礎ステージでローコードデジタルファクトリーをセットアップすることで、STEP（構造、人材、エコシステム、プロセス）の各次元が組み込まれます。短期間で成果を得るための準備ができるよう、これらの次元を順を追って説明していきます。ここで説明する方法は、デジタルオペレーションとデジタルエクスペリエンスのユースケースを念頭に置いています。これは、基礎ステージが両方に適用可能であるためです。



構造

“プロットとは、ストーリーの中の重要な出来事をまとめたものです。人体に骨が必要であるように、ストーリーには構造が必要です。どのように構造に「肉付けし、衣を着せる」かによって、それぞれのストーリーに独自性が生まれます。”

ー キャロライン・ローレンス（作家）

ローコードを使用してデジタルトランスフォーメーションを高速化することは、ソフトウェア構築の既存のアプローチに影響を与えます。ローコードデジタルファクトリーは、ローコードのアプリケーション開発/デリバリーをサポートする上で最適な組織構造を提供します。しかし、プロジェクトを担う理想的な開発者を探す前に、まずは導入チームを立ち上げる必要があります。チームには、推進者として熱意を持ってプロジェクトを継続する役割を担う企業トップからの支援、そして過去に導入に携わったことのあるエキスパートを含めます。

導入チーム

変化は自然に起こるものではありません。それだけでありません。以前と同じやりかたが通じなくなるために戸惑いも生じます。したがって、組織全体で変革を推進する力が必要となります（ボールを坂の上へ転がすには押す必要があるのと同じです）。この推進力（導入チーム）がなければ、わずかな失敗さえもプログラム全体を反故にしかねません。

導入チームは、いくつかの重要な役割で構成されます。



エグゼクティブスポンサー

組織を変更するには、企業トップの支援が必要です。説得力のあるビジョンを明確に示し、必要な資金を調達できるエグゼクティブスポンサーは、変革のためのあらゆるプログラムで欠かせません。

導入推進者

デジタルトランスフォーメーションの取り組みを映画にたとえるなら、導入推進者はヒーローであり、プログラムの成功とコンセンサスの形成を促進します。必要とされる特性は熱意と打たれ強さです。

導入エキスパート（OutSystemsまたはOutSystemsパートナー）

ヒーローは、過去に同じ経験をしており、貴重な助言と方向性を示す師範からの手ほどきを必要とします。導入推進者に導入エキスパートの経験とサポートを連携させることで、不測の事態を回避できます。

幸いなことに、このボールは一度押されると勝手に転がり始め、加速していきます。

これが変革の本質です。つまり、組織全体でローコードを導入するには、最初に集中的に取り組むチームが必要となります。

最小のローコードデジタルファクトリー

すべての組織の構造は異なりますが、ソフトウェアデリバリーには共通の機能があります。これらの機能は、6つの領域に分類できます。基礎ステージでのアプリケーションの構築には1つのチームとして関与するので、ローコードデジタルファクトリーには簡単に着手できます。また、この図に示すような提案に従うことで、最初のアプリケーションデリバリーを確実に成功させることができます。



ビジネス
ユーザー



プロダクト
オーナー



テックリード



開発者



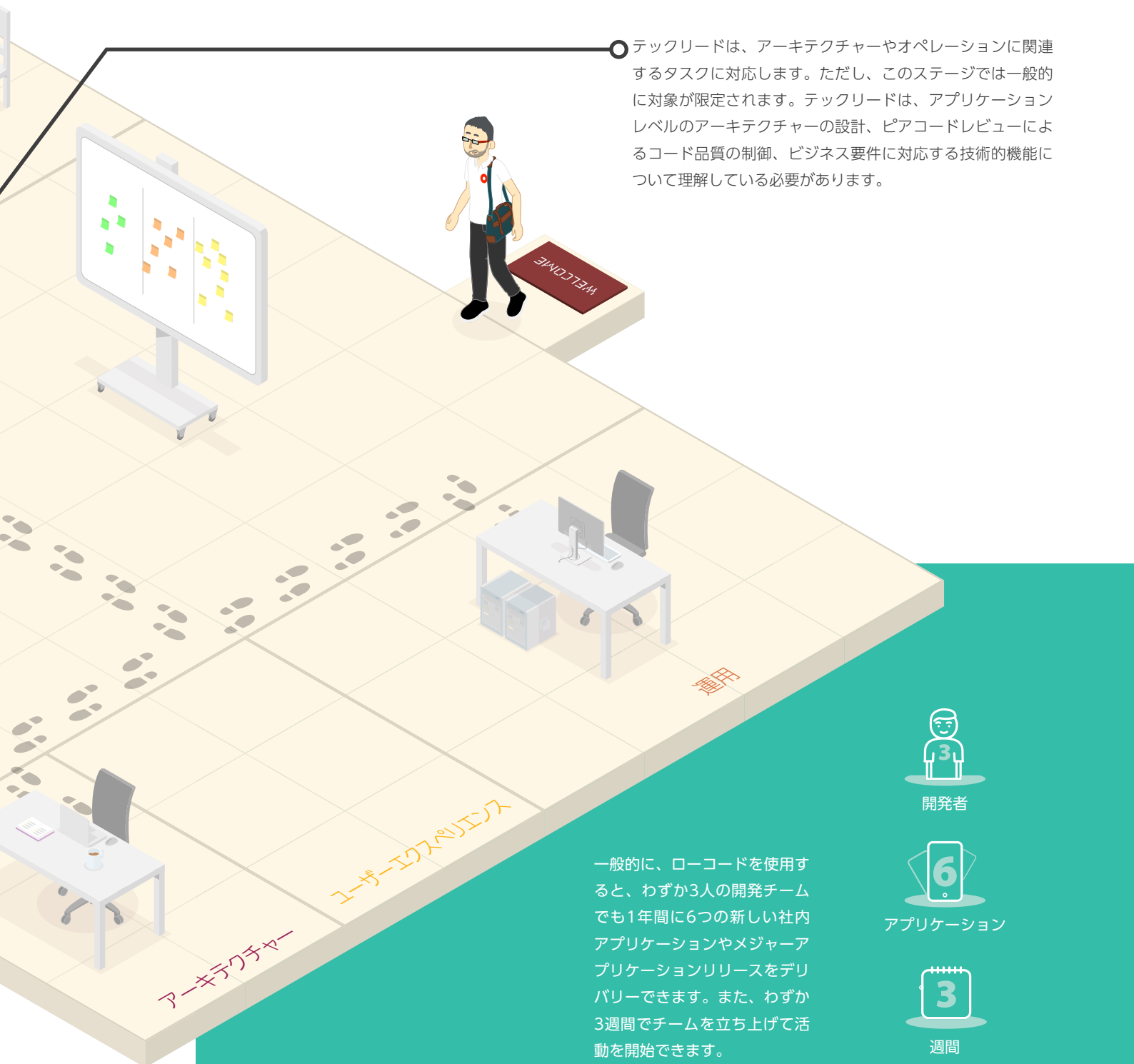
導入推進者



導入
エキスパート

- プロダクトオーナーは、開発チームとビジネスユーザーの間の代理人として機能します。ただし、通常、このステージでは正式なプロジェクト管理手法は不要です。その代わりに、ビジネスユーザーが開発チームに直接関与する可能性が高くなります。

- テックリードは、アーキテクチャーやオペレーションに関連するタスクに対応します。ただし、このステージでは一般的に対象が限定されます。テックリードは、アプリケーションレベルのアーキテクチャーの設計、ピアコードレビューによるコード品質の制御、ビジネス要件に対応する技術的機能について理解している必要があります。



開発者



アプリケーション



週間

デジタルエクスペリエンスのための構造の調整

デジタルエクスペリエンスのユースケースについては、基礎ステージで追加のリソースが必要となります。この時点で、チームにさらに役割を追加します。追加する役割は、アプリケーションが社内のユーザーや消費者にとって直感的で使いやすく、歓迎されるものであることを確認するためのものです。

顧客中心の考え方

顧客に注力するには、その考え方を理解する必要があります。主要ユーザーを特定し、設計と開発のプロセス全体でチームに参加してもらい、新しいソリューションのデリバリーでコメントや意見を仰ぎます。

テストの役割の拡張

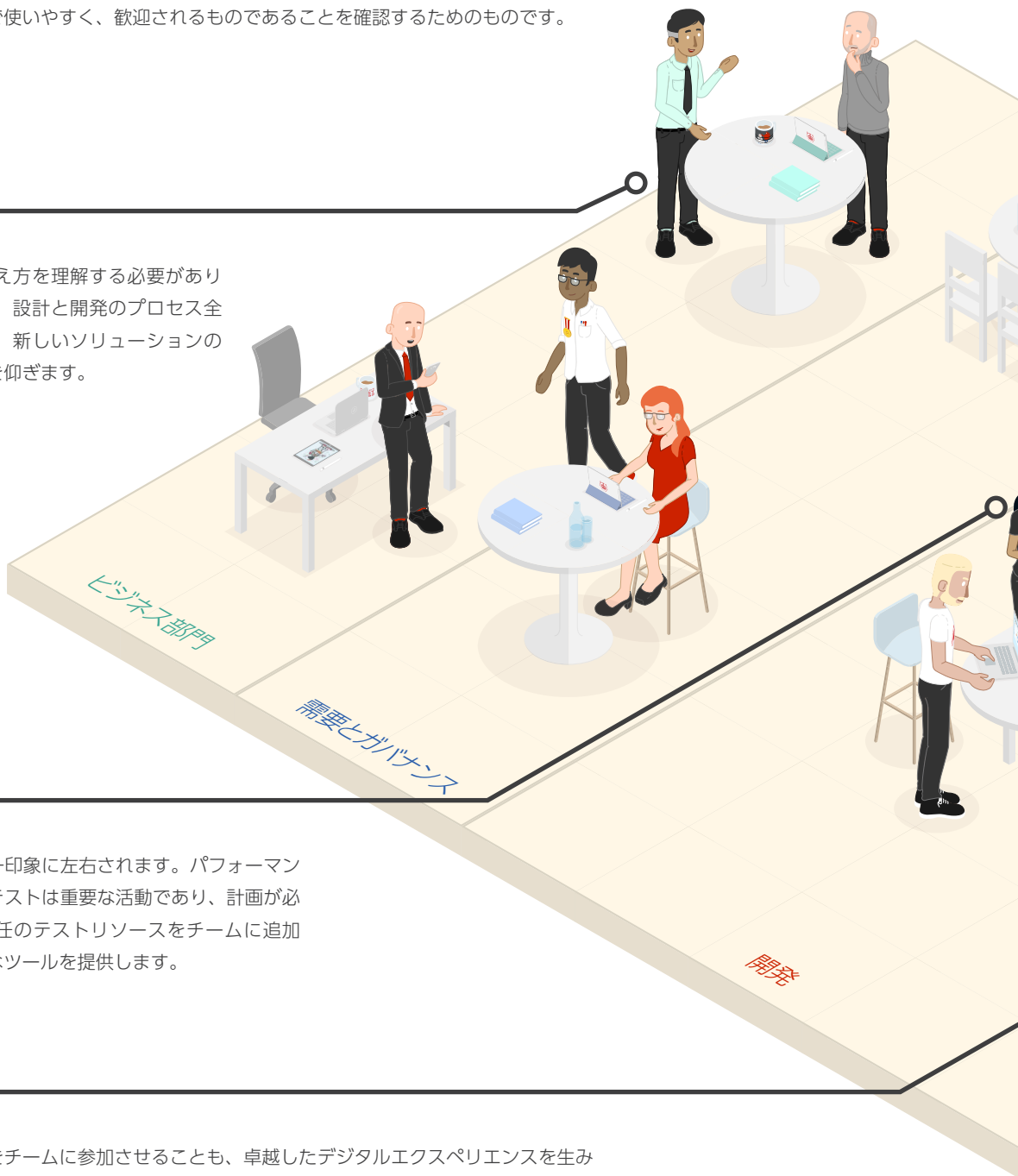
アプリケーションの成否は第一印象に左右されます。パフォーマンステストとデバイス横断的なテストは重要な活動であり、計画が必要となります。知識豊富な専任のテストリソースをチームに追加し、確実な成功のために適切なツールを提供します。

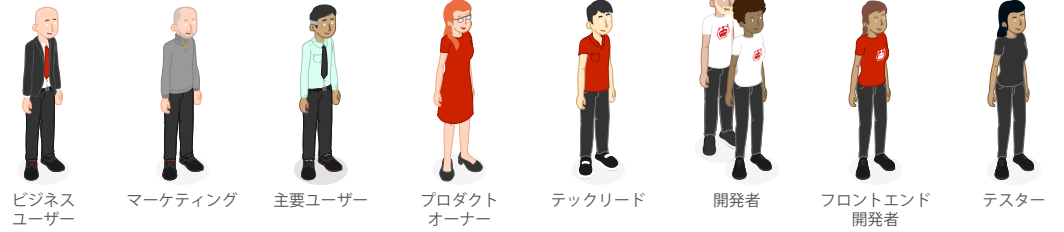
フロントエンド開発者

フロントエンドの専任開発者をチームに参加させることも、卓越したデジタルエクスペリエンスを生み出すプロセスにプラスの効果生まれます。ローコードプラットフォームにより、すべてのチームメンバーが洗練されたユーザーエクスペリエンスを非常に簡単に構築できるようになります。しかし、このプラットフォームによってその意欲が生まれるわけではありません。ユーザーやユーザーエクスペリエンスへの入念な注力、およびプロセスの一部となる細部へのこだわりは非常に重要となります。

UX/UIデザイナー

カスタマージャーニーを理解し、最適化された経験を開発するには、専門のスキルが必要です。UX/UIデザイナーをチームに加えることで、顧客のニーズが考慮されるようになり、期待以上の結果を実現できます。





第1級のサポート

顧客サービスチームは、ユーザーとのコミュニケーションを担い、新しい顧客ソリューションを提供するときに、そのソリューションの一部として優れたサポートサービスを提供できるように準備する必要があります。場合によっては、新しいデジタルエクスペリエンスをサポートするための別のデジタルオペレーションアプリケーションを作成する必要があります。

人材

“人材はCEOの最優先事項です。ビジョンや戦略に気を取られがちですが、適切な人材の確保が先決です。”

— アンドレア・ジュング（グラミン・アメリカ社長兼CEO）

ローコードの市場は劇的な勢いで拡大してきました。しかし、従来のコーディング手法と比較すると、その理解はいまだに進んでいません。適切な人材を見つけることは（社内調達する場合も、市場から雇用する場合も）、必要なスキルについての理解を深めない限り困難です。どのような要素に注目すべきかを理解できるよう、ローコードに関連する主な役割のプロファイルを紹介합니다。このセクションで使用する各コンピテンシー領域とスキルレベルの解説については、付録を参照してください。

基礎レベルのリソース

ローコードの取り組みを始めるために必要なスキルは、現在のチーム内にすでに存在する可能性があります。以下に示すプロファイルは、能力とスキルレベルをチームメンバーに対応させるための指針となります。さらに、一般的にローコードプラットフォームのベンダーはオンライン学習資料を提供しているので、これにより数週間でチームの活動を加速させることができます。また、必要なサポートを受けるためのプログラムも利用できます。これらを活用することで（しかも、ローコードテクノロジーは短時間で学習できます）、多くの場合にチームは数週間でビジネス価値を提供できるようになります。

開発者



分析と設計	■■■■■
Webアプリケーションのアーキテクチャー	■■■■■
ユーザーエクスペリエンス	■■■■■
SQLプログラミング	■■■■■
開発言語	■■■■■
UI開発言語	■■■■■
統合	■■■■■
パフォーマンス	■■■■■
トラブルシューティングとデバッグ	■■■■■

開発者は、経験豊富なテックリードの監督下で活動し、ビジネスソリューションを提供するコラボレーション型アプリケーションを設計、構築、テストします。

市場でターゲットにすべき人材は、.Net、Java、Javascriptを使用するソフトウェア開発で3年以上の経験を持つ開発者です。

この役割は、革新的なWeb/モバイルのビジネスアプリケーションを開発するために、最新のソフトウェアエンジニアリングプラクティスと組み合わせたOutSystemsテクノロジーを習得します。

テックリード



分析と設計	■■■■■
Webアプリケーションのアーキテクチャー	■■■■■
ユーザーエクスペリエンス	■■■■■
SQLプログラミング	■■■■■
開発言語	■■■■■
UI開発言語	■■■■■
統合	■■■■■
パフォーマンス	■■■■■
トラブルシューティングとデバッグ	■■■■■

テックリードは、複雑な技術的問題やビジネス課題を解決する革新的なソリューションの実装を推進します。テックリードは、技術チームとビジネスチームの連携を確立するためにアジャイルのアプローチを使用します。

ターゲットにすべき人材は、チームリーダーの役割と顧客対応で2年以上の経験を持つ専門家です。

経験豊富な開発者とアーキテクチャーチームをすでに配置している場合は、社内でのプロモーションを検討します。市場から採用する場合は、リーダーシップの機会を求めている上級開発者をターゲットにします。

デジタルエクスペリエンスのための人材の調整

デジタルエクスペリエンスの作成を開始すると、ユーザーエクスペリエンスやテストなど、他の役割がチームにとって重要になります。新しいデジタルエクスペリエンスを堅牢なものにし、顧客に好印象を与えるには、専門のスキルが必要とされます。組織内の人材やチームメンバーが、すでにこれらのスキルを持っている可能性があり、それらのスキルをローコードに移転できます。ただし、安易にチーム内の開発者に複数の役割を負わせないように注意してください。追加の役割は、担当するデジタルエクスペリエンスのタスクに完全に注力すべきであり、適切な経歴を持つリソースが対応するのが最適です。必要とされるプロフィールは以下のとおりです。

UX/UI デザイナー



ユーザー調査	<div style="width: 25%;"></div>
情報アーキテクチャー	<div style="width: 25%;"></div>
低品質のモックアップ	<div style="width: 25%;"></div>
ビジュアルデザイン	<div style="width: 25%;"></div>
ユーザーテスト	<div style="width: 25%;"></div>
顧客管理	<div style="width: 25%;"></div>
モバイル	<div style="width: 25%;"></div>
ユーザビリティ	<div style="width: 25%;"></div>
ビジネスプロセス	<div style="width: 25%;"></div>
カスタマージャーニー	<div style="width: 25%;"></div>

UX/UIデザイナーは、ユーザーエクスペリエンスのルックアンドフィールを視覚的に示すクリック可能なプロトタイプを提供することで、アプリケーションの価値に対する理解を深めてくれます。最も重要なのは、これがカスタマージャーニーのマッピングを担う役割である点です。

一般的に、適任のUX/UIデザイナーは、分析的な考え方の持ち主で、物事を現実的に捉えることができる人材です。この役割の活動は質問を駆使した調査を伴います。したがって、UI/UX分野の経歴を持つデザイナーだけでなく、心理学やジャーナリズムの学位を持つ人材も候補者として検討できます。

フロントエンド 開発者



分析と設計	<div style="width: 25%;"></div>
Webアプリケーションのアーキテクチャー	<div style="width: 25%;"></div>
ユーザーエクスペリエンス	<div style="width: 25%;"></div>
SQLプログラミング	<div style="width: 25%;"></div>
開発言語	<div style="width: 25%;"></div>
UI開発言語	<div style="width: 25%;"></div>
統合	<div style="width: 25%;"></div>
パフォーマンス	<div style="width: 25%;"></div>
トラブルシューティングとデバッグ	<div style="width: 25%;"></div>

フロントエンド開発者は、ユーザーインターフェイスを実装し、ユーザーエンゲージメントのためのインタラクティブな要素をアプリケーションに追加する役割を担います。スケーラブルなフロントエンドを構築する経験を持っている必要があります。

ターゲットは、HTML、CSS、JavaScript、およびレスポンシブWebデザインのスキルを備えた経験豊富なフロントエンド/UI開発者であり、デジタルエクスペリエンスのテクノロジーツールに精通し、開発者と効果的にコミュニケーションをとることができる必要があります。

テスター



分析と設計	<div style="width: 25%;"></div>
Webアプリケーションのアーキテクチャー	<div style="width: 25%;"></div>
ユーザーエクスペリエンス	<div style="width: 25%;"></div>
SQLプログラミング	<div style="width: 25%;"></div>
開発言語	<div style="width: 25%;"></div>
UI開発言語	<div style="width: 25%;"></div>
統合	<div style="width: 25%;"></div>
パフォーマンス	<div style="width: 25%;"></div>
トラブルシューティングとデバッグ	<div style="width: 25%;"></div>

テスターは、テスト戦略を定義し、テストケースを作成し、自動/手動の手法を使用してアプリケーションをテストします。

ターゲットは、ソフトウェアQA、テストのプランニング、テスト作成で3年以上の経験がある専門家です。さらに、テスト自動化で1年以上の経験があり、継続的インテグレーションのテストフレー

ムワーク (Selenium, Robot, Jenkins, Apache JMeterなど) を理解し、デバイスファームの知識を有することが期待されます。

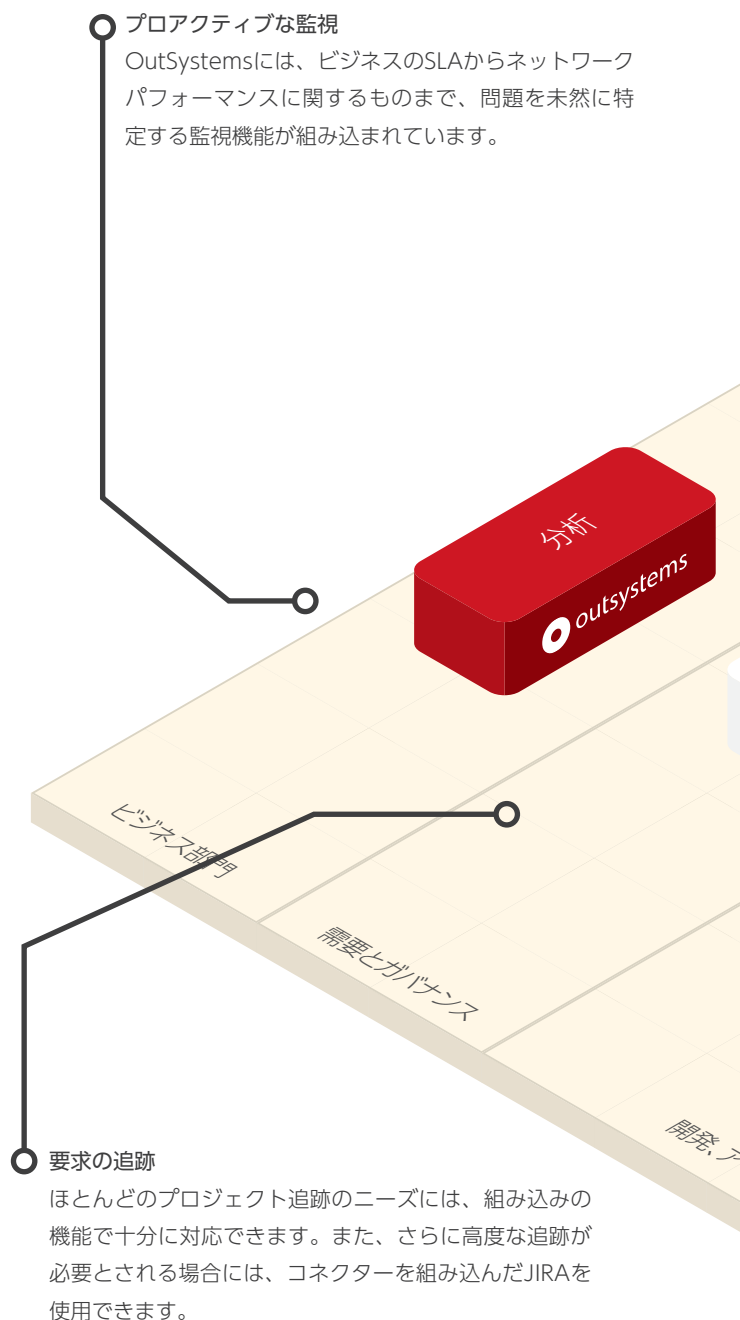
適任のテスターは、好奇心があり、細部に気を配り、物事をうのみにせず、顧客中心の考え方ができ、高い対人能力を備えている必要があります。

エコシステム

テクノロジーの変化は、決して孤立した現象ではありません。この変革は、企業、政府、社会の各側面から成る複雑なエコシステムの中で起こっています。イノベーションが推進する新たな競争に国家が適合するには、エコシステム全体を考慮しなければなりません。”

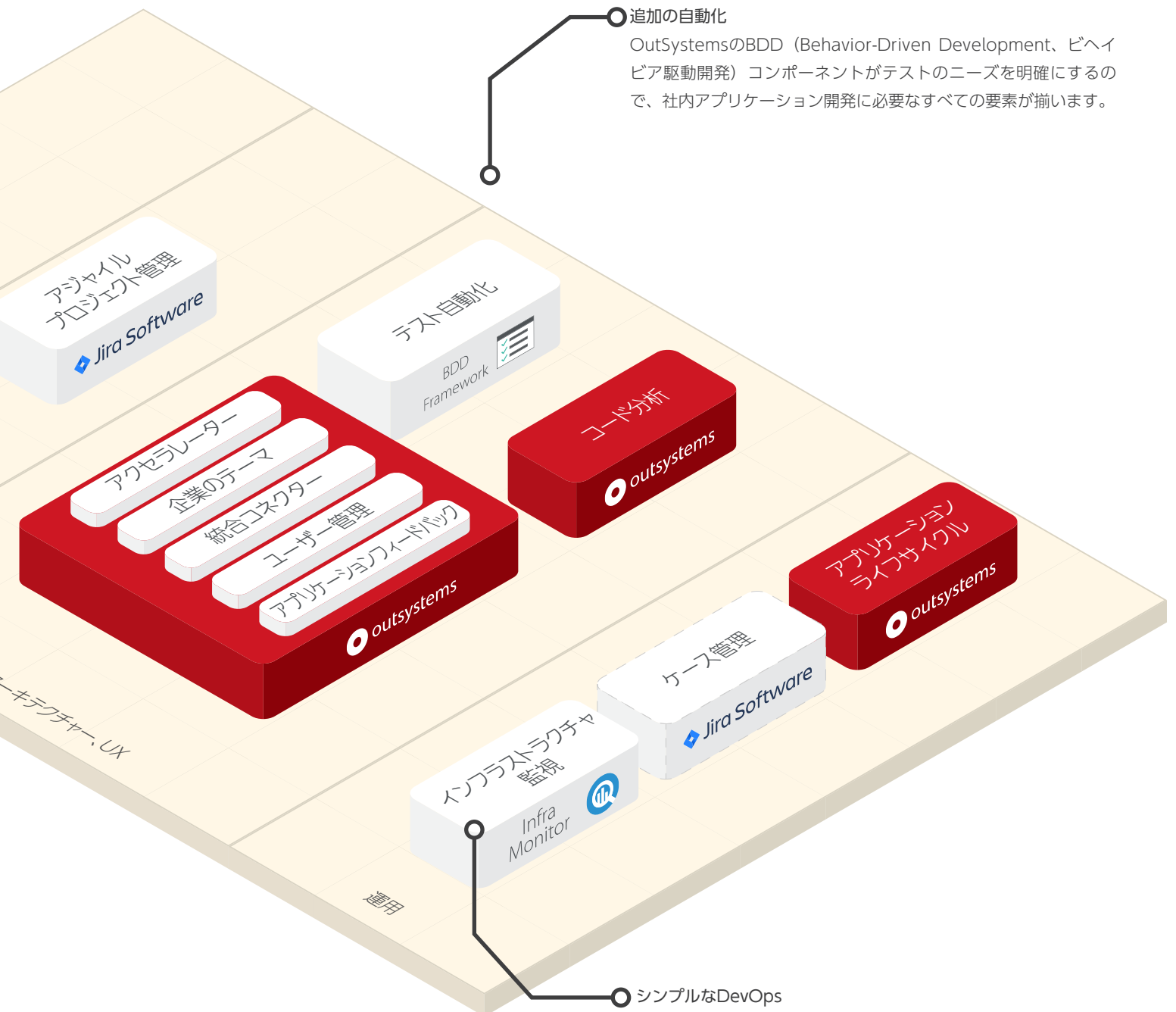
ー クラウス・シュワブ (世界経済フォーラム会長)

新しいテクノロジーの導入は、既存のツールのエコシステムにいかに対応するかによって成否が分かります。画期的なソリューションであっても、社内の主要システムに統合されなければ、広範に使用されたり頻繁に使用されたりすることはありません。せっかくのソリューションが活用されずに終わるのを回避するには、慎重な戦略が必要です。しかし、組織の構造と同様に、小規模な活動として開始することで、ソリューションの価値を引き出すのにかかる時間を短縮できます。



最小のローコードエコシステム

この図は、OutSystemsを中心とするエコシステムにおける基礎ステージでの依存関係を示しています。この単純なアプローチでは、これらのツールがいずれもOutSystemsと統合され、ダウンロードして即座に利用できます。

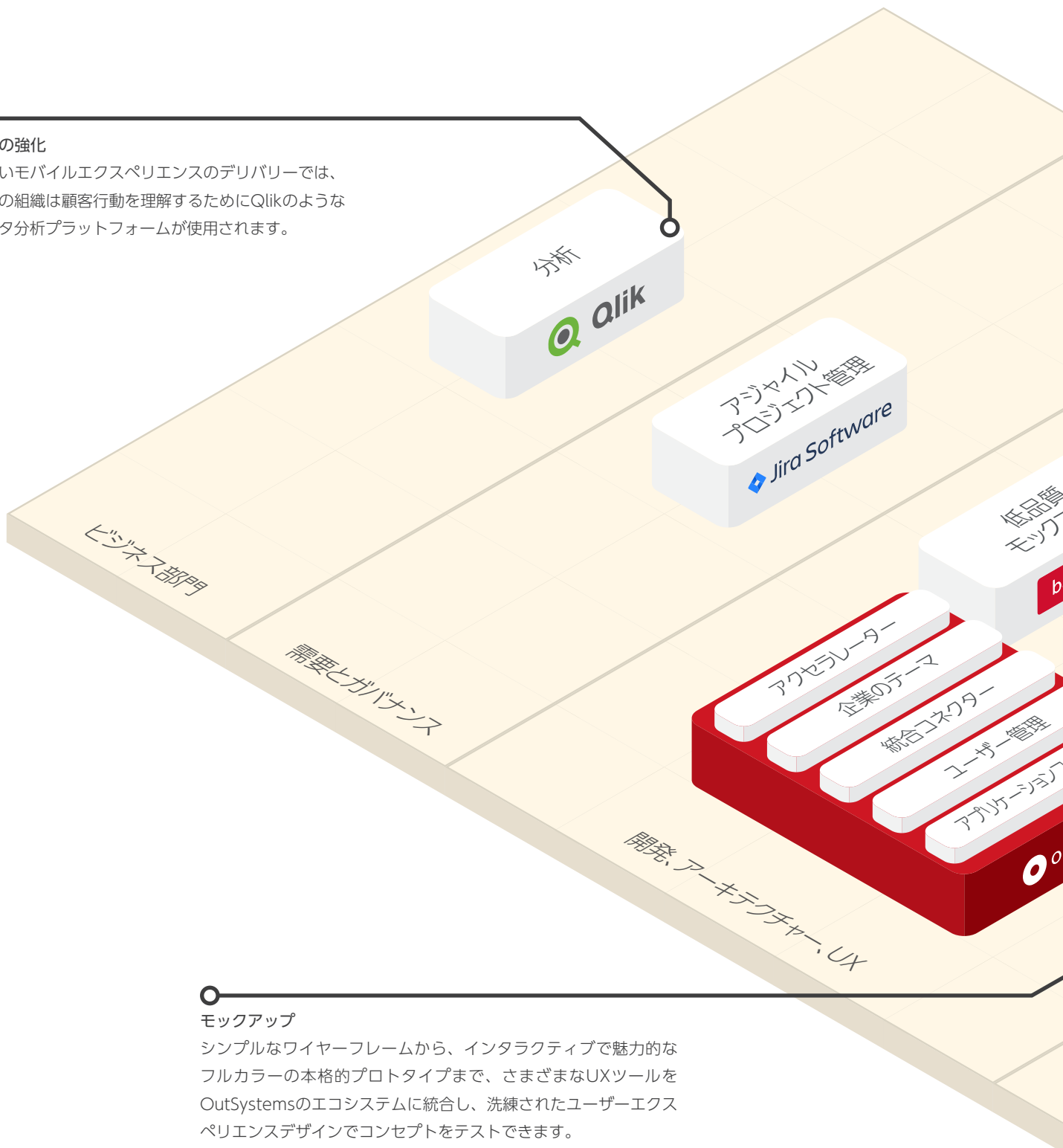


デジタルエクスペリエンスのためのエコシステムの拡張

依存関係は、ユーザーエクスペリエンスに対処するために拡張されます。この図は、幅広いお客様の成功に基づいて拡張されたエコシステムのツールを示しています。OutSystemsは、これらのツール向けに統合機能やコネクターを提供しています。

分析の強化

新しいモバイルエクスペリエンスのデリバリーでは、多くの組織は顧客行動を理解するためにQlikのようなデータ分析プラットフォームが使用されます。



モックアップ

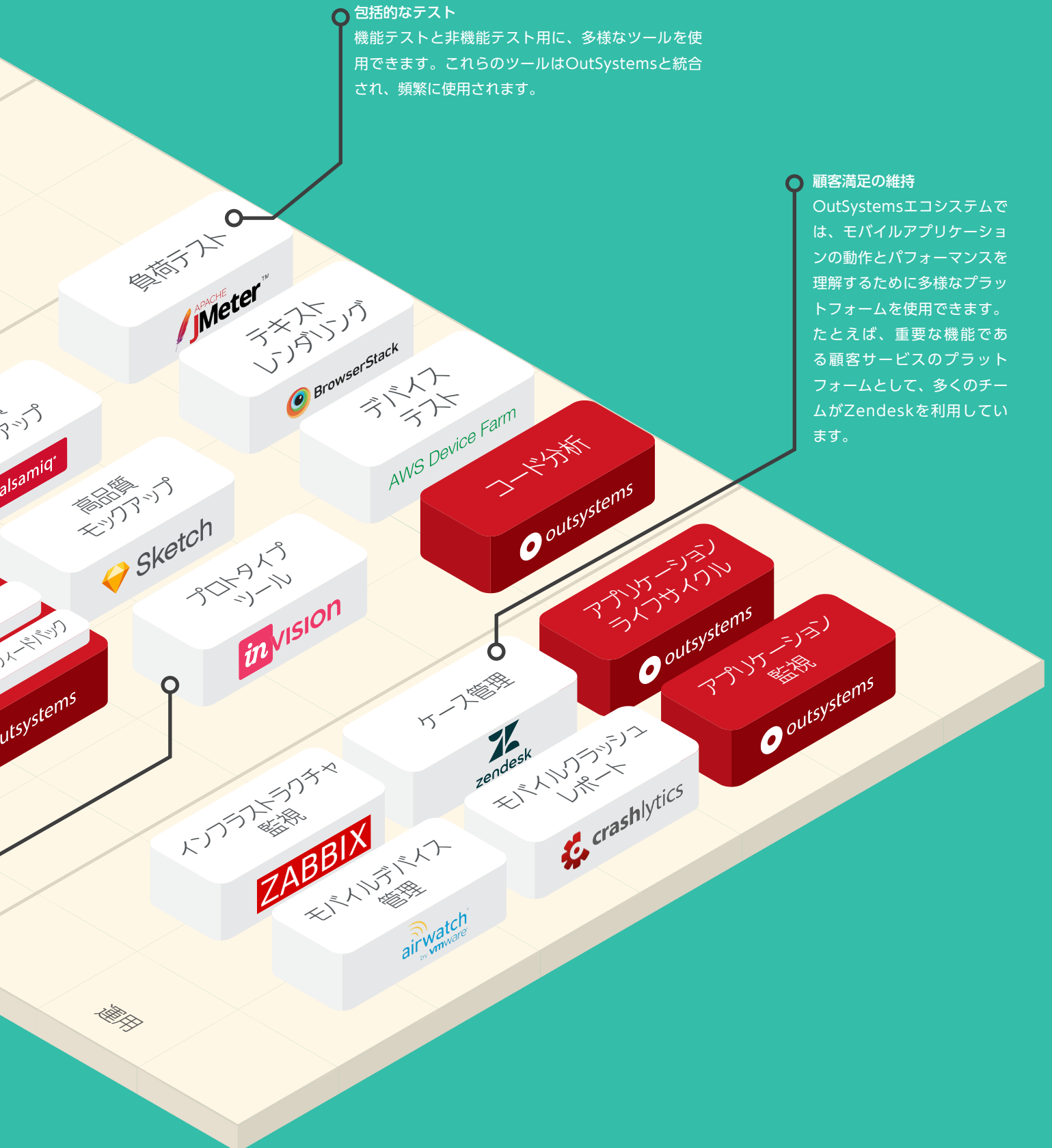
シンプルなワイヤーフレームから、インタラクティブで魅力的なフルカラーの本格的プロトタイプまで、さまざまなUXツールをOutSystemsのエコシステムに統合し、洗練されたユーザーエクスペリエンスデザインでコンセプトをテストできます。

包括的なテスト

機能テストと非機能テスト用に、多様なツールを使用できます。これらのツールはOutSystemsと統合され、頻繁に使用されます。

顧客満足度の維持

OutSystemsエコシステムでは、モバイルアプリケーションの動作とパフォーマンスを理解するために多様なプラットフォームを使用できます。たとえば、重要な機能である顧客サービスのプラットフォームとして、多くのチームがZendeskを利用しています。



プロセス

本来するべきでないことを効率的に行うほど無駄なことはありません。”

—ピーター・F・ドラッカー（マネジメント発明者）

ローコードを使用するデリバリーは、従来の手作業のコーディングとは異なります。したがって、適切なタイミングで適切な作業を実行するようにチームが注力するには、異なるプロセスが必要です。運用可能なアプリケーションデリバリーのために最適化されたプロセスフローから開始します。これは4つのカテゴリに分類できます。それぞれの活動について、タスクの遂行に責任を負う役割、タスクの完了に対する最終的な説明責任を負う役割、活動中に助言を与える役割も示されています。付録では、フロー内のそれぞれの活動の詳細を説明しています。

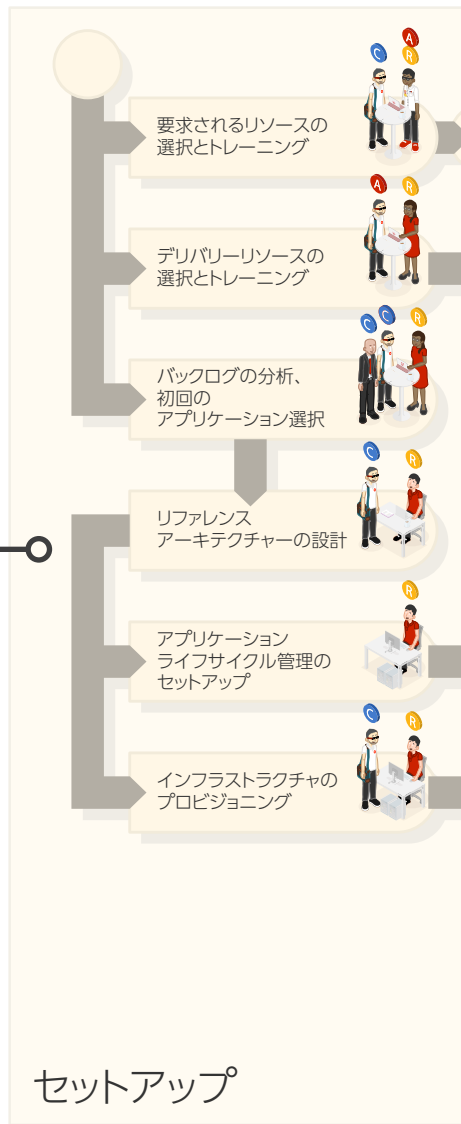


要求

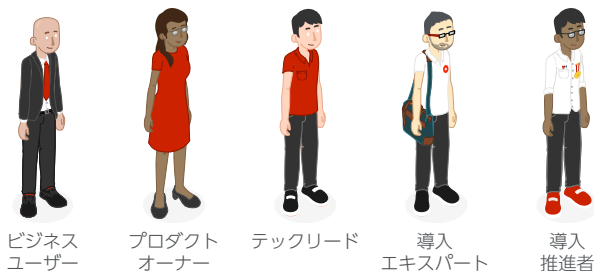
プロジェクト管理の基礎レベルでは、正式な活動はあまり多くありません。しかし、新しいソリューションの価値を検討し、アプリケーションの構築自体が目的にならないように注意する必要があります。アイディエーションの活動では、最も価値の高い機能を特定し、アプリケーションの構築に必要な労力とアプリケーションがもたらす価値を比較します。

セットアップ

基礎レベルでは、セットアップの要件はわずか数週間で完了できます。チームの準備、開発するアプリケーションの選択、アーキテクチャが十分であることの確認、リリースパイプラインのセットアップを中心に行います。



- R** タスク遂行の責任
- A** タスク完了の最終的説明責任
- C** 助言者
- i** 情報提供

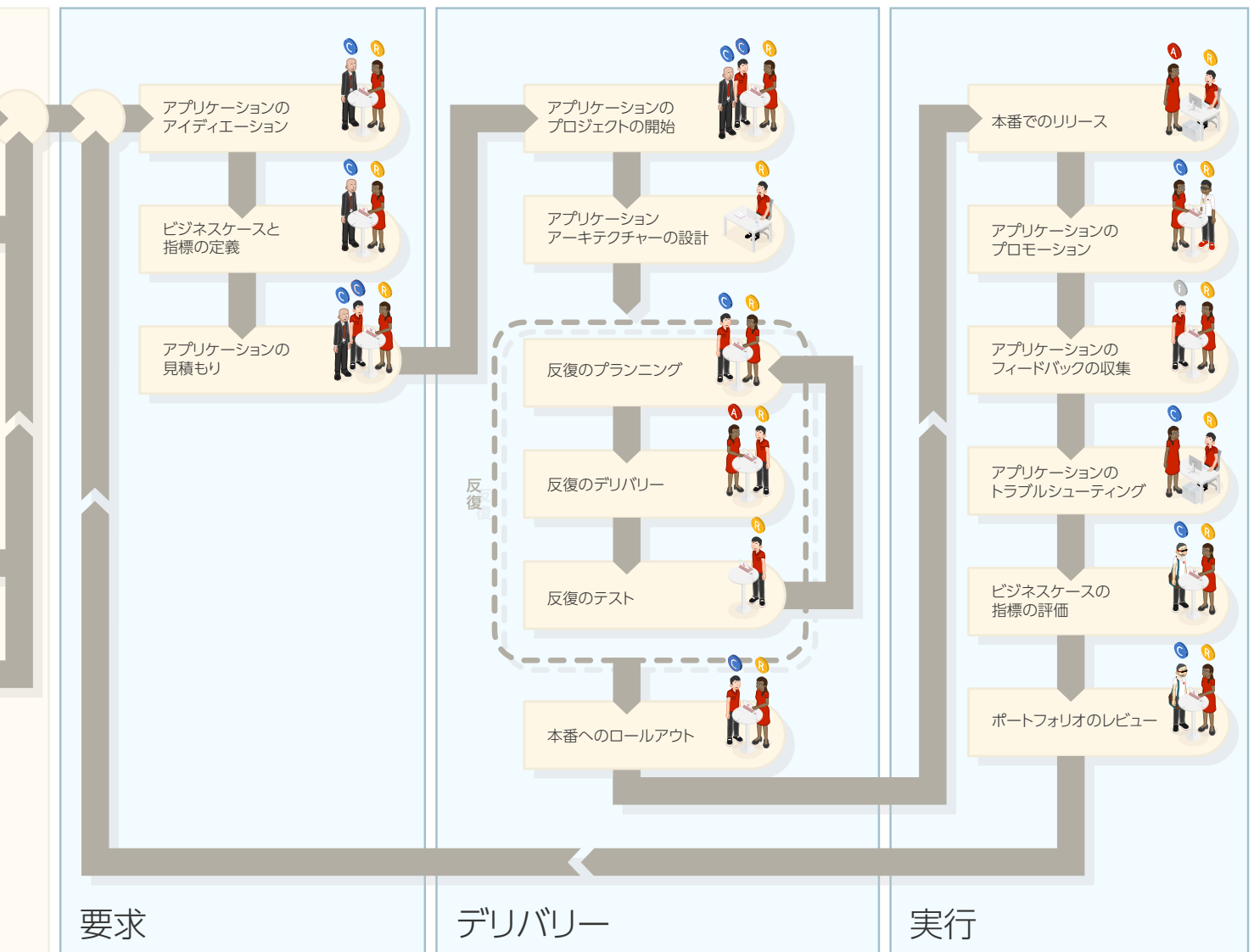


デリバリー

アプリケーションのすべてのレイヤーを一貫したビジュアルアプローチで開発し、展開をワンクリック操作に簡素化することで、期待される機能横断的なアジャイルチームを実現できます。

稼働

本番稼働へのアプリケーションの投入は一里塚になりますが、そこで活動が終わるわけではありません。アプリケーションのプロモーション、フィードバックの収集、トラブルシューティング、ビジネスへの影響の評価は、アプリケーションを成功させるための重要な要素です。

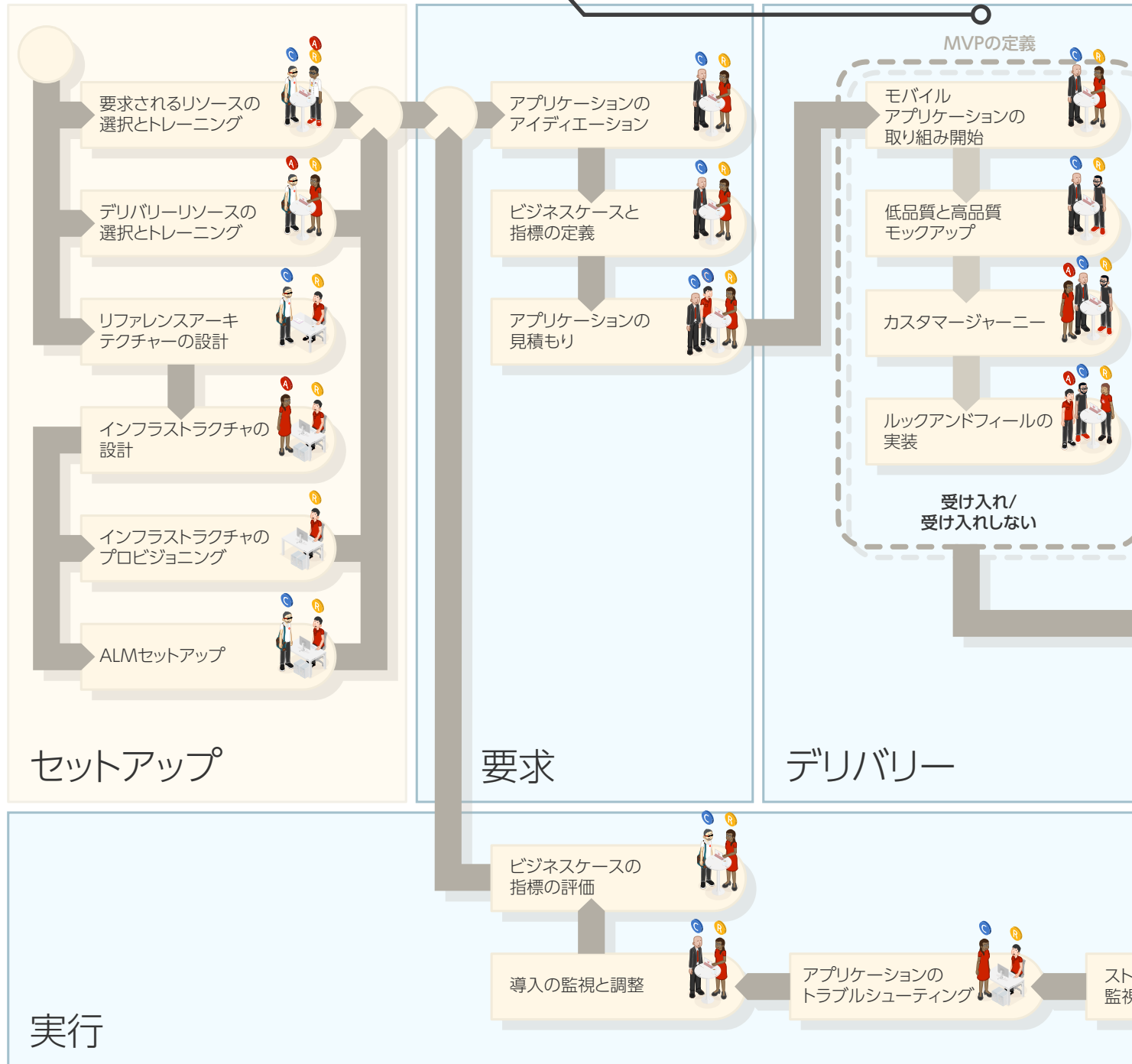


デジタルエクスペリエンスのための調整

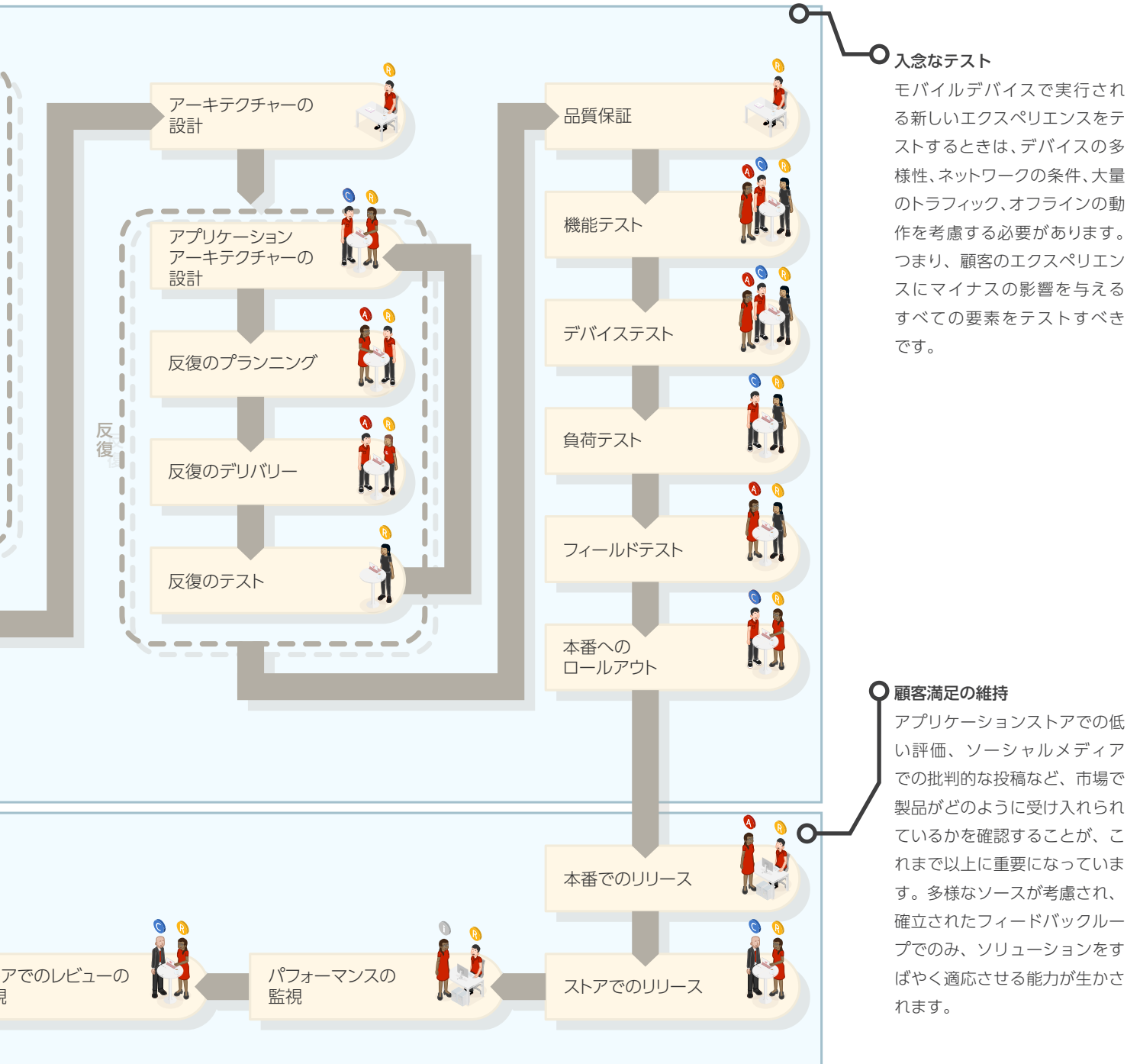
デジタルエクスペリエンスのユースケースの基礎レベルでは、プロセスフローが変化し、ユーザーを満足させるための活動が追加されます。

MVPの開発

リーンスタートアップのコンセプトは、出発点としてMVP（実用最小限の製品）を提供することです。これは、新しいデジタルエクスペリエンスのために重要です。検証と早期フィードバックにより、MVPの進化が規定されます。その結果、より正確かつ効果的に顧客、従業員、またはパートナーをターゲットとする最終製品を実現できます。



-  タスク遂行の責任
-  タスク完了の最終的説明責任
-  助言者
-  情報提供



入念なテスト

モバイルデバイスで実行される新しいエクスペリエンスをテストするときは、デバイスの多様性、ネットワークの条件、大量のトラフィック、オフラインの動作を考慮する必要があります。つまり、顧客のエクスペリエンスにマイナスの影響を与えるすべての要素をテストすべきです。

顧客満足の維持

アプリケーションストアでの低い評価、ソーシャルメディアでの批判的な投稿など、市場で製品がどのように受け入れられているかを確認することが、これまで以上に重要になっています。多様なソースが考慮され、確立されたフィードバックループでのみ、ソリューションをすばやく適応させる能力が活かされます。

まとめ

デジタルトランスフォーメーションやローコードプラットフォームに関する議論では、「それが自社の役に立つのか？」という質問が必ず出されます。これに答える最善の方法は、できるだけ早く価値を実証することです。

ローコードを導入した多くの企業では、次のような発言が聞かれます。「小規模なプロジェクトから始め、わずかな時間で実用的なアプリケーションを実現できたことに驚きました。ローコードの可能性を確認できたので、さらに大規模なプロジェクトに取り組むことを決定しました」これらの企業は、当初はローコードに懐疑的であったものの、長期的にインパクトのある成功が迅速にもたらされたことから納得した人々の例を挙げています。また、以前は6か月かかっていた重要アプリケーションの開発をわずか6週間で達成したという、忘れられない経験も語られています。

基礎ステージは、当社が把握しているすべての成功事例に基づいています。つまり、単純さを維持し、小規模なものから始めることで、誰でもプロジェクトを迅速かつ簡単に遂行できるのです。したがって、ローコードデジタルファクトリーをセットアップして、このステージに取り組むことで、組織が他のステージに移行しても、いつでも適応できます。言い換えれば、オイルを十分に差した機械のようにプロジェクトを順調に開始して、数週間後には本番稼働させることができるのです。

これが、「自社の役に立つのか？」という質問に対する説得力のある回答になります。

次のステップ

ローコードデジタルファクトリーによりサポートされるデジタルトランスフォーメーションの取り組みは、始まったばかりです。次のプレイブックでは、3つのユースケースのセンターオブエクセレンスの確立と、このステップへのローコードデジタルファクトリーの適応について取り上げます。





付録

コンピテンシーのスキルレベルの内訳

人材のセクションに記載されているローコードの各役割については、必要なコンピテンシーが以下の表に従って5段階で評価されます。

スキルレベル：0 | 適用せず

このコンピテンシーは役割に適さない。

スキルレベル：1 | 基礎知識

一般的知識または基本的な手法や概念の理解がある。

注力：学習。

スキルレベル：2 | 限られた経験

能力：このコンピテンシーに関連する用語、概念、原則、および課題を理解し、議論できる。

経験：クラスルームまたは実験シナリオ（あるいはその両方）、または現場のトレーニングで得た。

期待される成果：このスキルでは支援を必要とする。

注力：現場での経験を通じた開発。

スキルレベル：3 | 中級

能力：このコンピテンシーのタスクを成功裏に遂行できる。

経験：最小限のガイダンスにより、このコンピテンシーを状況に成功裏に適用したことがある。

期待される成果：独立してスキルを使用し、折に触れてさらに上級のリソースから支援を受ける。

注力：知識やスキルの適用と強化。

スキルレベル：4 | 熟練

能力：支援なしに、このスキルに関連するタスクを遂行できる。

経験：プロセスやプラクティスに関する容易に実現可能な改善について、実践的で関連度の高いアイデアや視点を継続的に提供している。

期待される成果：このコンピテンシーの適用で、複雑なニュアンスをわかりやすい言葉に置き換えて他者を指導する。

注力：組織的または専門的な幅広い課題。

スキルレベル：5 | 上級

能力：この専門分野またはスキルが使用される分野に関連して、ガイダンスを提供し、トラブルシューティングを行い、質問に回答できる。

経験：複数のプロジェクトや組織にわたるコンピテンシーの適用で、一貫して優位性を実証している。また、このコンピテンシーの新しい適用先を作成し、このコンピテンシー向けの参照/リソース資料の開発を主導したことがある。

期待される成果：この領域で、組織内外から「頼りになる」人物として認知される。

注力：戦略的。

コンピテンシーの内訳

次の表は、人材の次元の一部として特定された9つのコンピテンシー領域について詳細を示しています。

スキル	説明
分析と設計	
アーキテクチャー/ ソリューション設計	<ul style="list-style-type: none"> 適切なアーキテクチャー原則を習得しており、4層のアーキテクチャーを使用してアプリケーションを設計できる 優れたパフォーマンス、スケーラビリティ、および保守性を保証する 循環参照を回避、検出、修正する方法を理解している ツールを使用してプロジェクトのアーキテクチャーを監視する方法を理解している
要件収集	<ul style="list-style-type: none"> 要件収集の手法を習得している： <ul style="list-style-type: none"> ビジョンと推進要素を特定する ビジネスのコンテキストを明確に把握する ペルソナを特定する ユーザーストーリーを定義する ビジネスプロセスモデルを開発する モックアップを構築する
Webアプリケーションのアーキテクチャー	
Webアプリケーションの アーキテクチャー	<ul style="list-style-type: none"> Webアプリケーションのアーキテクチャーを理解している (クライアント/サーバーアーキテクチャー)： <ul style="list-style-type: none"> POSTとGETをいつどのように実行するか ページレンダリング データベースアクセス セッションの動作を把握する ブラウザのキャッシュを把握する 状態を確認する Ajaxリクエスト
ユーザーエクスペリエンス	
ユーザビリティを理解し、 適用する	<ul style="list-style-type: none"> 情報アーキテクチャーを理解している： <ul style="list-style-type: none"> それが何であるか なぜ構築するか 編成 ラベル付け コスト ナビゲーション ユーザーの場所 補助的なナビゲーション デザイン原則のSCRAP (対称、コントラスト、繰り返し、アライメント、近接)を理解し、アプリケーションの構築時に適用できるようにする IT開発者のための11のUXルールを理解している ユーザビリティテストの重要性を理解し、実行できる 実際のサンプルデータを使用する、または現実的なテストデータを生成する

分析と設計

SQLプログラミング

- 以下を実装できる：
 - 一般的なテーブル結合 (LEFT JOIN、INNER JOIN)
 - EXISTS
 - COUNT
 - GROUP BY
 - トップ
 - 行番号
 - 集計関数 (SUM、AVG、MAX、MIN)
- 以下を実装できる：
 - IN
 - BETWEEN
 - OUTER JOIN
 - SQL CASE
 - サブクエリ
- インデックスと、インデックスがパフォーマンスに及ぼす影響を理解している

トラブルシューティングとデバッグ

アプリケーションの
トラブルシューティング

- エラーログを分析できる
- 監視ログを分析できる
- パフォーマンスログを分析できる
- パフォーマンスの問題のトラブルシューティング、特定、解決の方法を理解している
- インデックスと実行計画を理解している
- パフォーマンスチューニングツールについて理解し、チューニングと監視のためにツールを使用する方法を理解している

開発言語

アルゴリズム、擬似コード

- 問題または要件を理解している
- ソリューション全体を考慮してデザインを開発する - ドリルダウンして説明できる

C#, Java, Python, PHP

- 言語のベストプラクティスと、最も一般的に使用されるコードライブラリについて理解している
- 基本的なアルゴリズムまたは機能を実装できる
- MS Visual Studio、Eclipseなど、IDEの経験がある

iOS、Android

- Swift、Java

UI開発言語

CSS

- CSS構文について理解している
- ボックスモデルについて説明できる
- 疑似要素について理解している
- ブラウザーの問題のトラブルシューティングを実行できる
- OutSystemsでのCSSのインポート順序を理解している
- セレクターを読んで適用する方法を理解している
- CSSとCSS3の違いを理解している
- 位置指定値の違いを理解している
- テーマの構成について理解できる
- 垂直方向のアライメントについて理解できる
- RESS (レスポンシブWebデザイン+サーバー側コンポーネント) とRWD (レスポンシブWebデザイン) のアプローチを説明できる

HTML	<ul style="list-style-type: none"> • どの要素がどのように使用されるかを理解している • HTML5で利用可能になったAPIについて理解している • HTMLの標準要素をHTMLファイルに追加する方法を理解している • アクセシビリティについて理解している
JavaScript	<ul style="list-style-type: none"> • JS/jQueryについて理解している • 要素をIDで選択する • 要素をクラスで選択する • 要素を表示したり非表示にしたりできる • jQueryプラグインをインポート（および使用）できる • ブラウザーの問題のトラブルシューティングを実行できる • JSオブジェクトを作成する方法を理解している • jQueryプラグインを作成する方法を理解している • jQuery Ajaxリクエストを手動で実行する方法を理解している • jQueryプラグインを作成し、コミュニティで共有したことがある
統合	
REST、SOAP	<ul style="list-style-type: none"> • REST APIとSOAP APIを使用できる • REST APIとSOAP APIを提供できる • RESTとSOAPの統合をテストする方法を理解している • SOAP APIのNull値を処理する方法を理解している
サードパーティとの統合	<ul style="list-style-type: none"> • SAP Wizard for Integration Studioを使用して拡張を作成する方法を理解している • OutSystemsとSAPの間でトランザクションを処理する方法を理解している
パフォーマンス	
アプリケーション パフォーマンスを評価する	<ul style="list-style-type: none"> • レポートをドリルダウンしてボトルネックを特定できる • パフォーマンスレポートを使用する（クライアント側の時間とサーバー側の時間の違いを理解している） • ブラウザーパフォーマンスツールを使用する
アプリケーションコードを 最適化する	<ul style="list-style-type: none"> • ページサイズを小さくする • ページリソースのサイズを小さくしてロードし、キャッシュする • データベースからフェッチしたデータを小さくする • 外部データソースからのデータをキャッシュする
アプリケーション インフラストラクチャを 最適化する	<ul style="list-style-type: none"> • データベースの劣化を回避する • 保守計画を使用する • データベースファイルの拡大を調整する • アプリケーションサーバーとWebサーバーのメモリ設定を改善する

作業項目の定義

セットアップ

活動	概要
プロダクトオーナーの 選択とトレーニング	<p>プロダクトオーナーはビジネスと対話し、ビジネスの要望とニーズを把握してユーザーストーリーにマッピングできなければなりません。POには熟練したプロジェクトマネージャーを任命し、プロセスの推進に責任を持ちます。ローコードの経験があれば、なお望ましいでしょう。</p> <p>タスク遂行の責任/タスク完了の最終的説明責任： 導入推進者</p> <p>助言者： 導入エキスパート</p>
デリバリーリソースの 選択とトレーニング	<p>テックリードと（最大3人の）開発者は、アプリケーション開発の経験があり、それぞれのスキルセットを持っている必要があります。OutSystemsの使用経験があれば、なお望ましいでしょう。テックリードは基礎レベルで他の機能を持ちますが、開発トレーニングを受ける必要があります。</p> <p>タスク遂行の責任： プロダクトオーナー</p> <p>タスク完了の最終的な説明責任： 導入エキスパート</p>
バックログの分析、 初回のアプリケーション選択	<p>ローコードの利点を実証し、ビジネス価値を早期に実現することでプラットフォームの導入に弾みをつけるため、最初のアプリケーションの選択が重要です。軽薄な根拠、理屈、または機嫌取りのために誤ったアプリケーションを最初に選択すると、取り組みの価値が認められずに放棄されかねません。このため、慎重に選択する必要があります。</p> <p>タスク遂行の責任： プロダクトオーナー</p> <p>助言者： ビジネスユーザー、導入エキスパート</p>
リファレンス アーキテクチャーの設計	<p>コストを抑えながら将来のアプリケーションを存続/進化させるよう導くリファレンスアーキテクチャーを設定するには、レガシーシステムとの統合のニーズ、認証/許可、監査証跡、スタイルガイドなどの一般的な非機能要件（NFR）を理解することが重要です。</p> <p>タスク遂行の責任： アーキテクトとして活動するテックリード</p> <p>助言者： 導入エキスパート</p>
インフラストラクチャーの プロビジョニング	<p>クラウドインフラストラクチャーでは、すべてが自動的にセットアップされ設定されます。クラウドのプロビジョニングにはDevOpsのセットアップ手順が追加的に必要になりますが、すべてが簡素化されます。</p> <p>タスク遂行の責任： DevOpsエンジニアとして活動するテックリード</p> <p>助言者： IT組織のインフラストラクチャー担当者。</p>
ALMのセットアップ	<p>プロジェクトの成功を保証し、アプリケーションライフサイクル全体にわたってスピードを維持できるように、ローコードアプリケーションライフサイクル管理の適切なプラクティスが不可欠です。異なるチームと役割をそれぞれのALMツールでセットアップする必要があり、展開パイプラインとリリース管理を定義する必要があります。</p> <p>タスク遂行の責任： DevOpsエンジニアとして活動するテックリード</p> <p>助言者： 導入エキスパート</p>

デジタルエクスペリエンスのための調整

活動	概要
インフラストラクチャの設計	<p>消費者向けのモバイルアプリケーションを開発する際は、いくつかの要件（アプリケーションを使用するユーザー数、冗長性、負荷分散など）に対応するようにインフラストラクチャを設計する必要があります。</p> <p>タスク遂行の責任： DevOpsエンジニアとして活動するテックリード</p> <p>助言者： プロダクトオーナー</p>

要求

活動	概要
アプリケーションのアイディエーション	<p>ビジネスの目標を達成するための革新的なソリューションを見つけます。潜在的な成果と障害を分析します。プロジェクトのデリバリーを成功させるには、要件と大まかな機能を特定する必要があります。</p> <p>タスク遂行の責任： プロダクトオーナー</p> <p>助言者： ビジネスユーザー</p>
ビジネスケースと指標の定義	<p>アイディエーションに基づいて、ビジネスケースと指標を定義できます。定義には、ビジネスの観点からのアプリケーションの意義、目標達成の評価方法などを含めます。経営陣が情報に基づいて意思決定を行うために、評価が必要とされます。</p> <p>タスク遂行の責任： プロダクトオーナー</p> <p>助言者： ビジネスユーザー</p>
アプリケーションの見積もり	<p>ビジネスケースを作成した後、アプリケーションの作成のために必要/利用可能な労力と予算を判断します。関連する要件と大まかな機能を検討します。</p> <p>タスク遂行の責任： プロダクトオーナー</p> <p>助言者： テックリード、ビジネスユーザー</p>

デリバリー

活動	概要
アプリケーションの着手	<p>アプリケーションの開発を始めるときは、初期のステップで開発フェーズのペースが決まります。最初の要件と大まかな機能を記録したら、開発の要件を定義してバックログを作成する準備が整います。</p> <p>タスク遂行の責任： プロダクトオーナー</p> <p>助言者： テックリード、ビジネスユーザー</p>
アプリケーションアーキテクチャーの設計	<p>アプリケーションを成功させるためには、スケーラビリティ、パフォーマンス、およびコンポーネントの再利用性が考慮された適切なアーキテクチャーが必要とされます。</p> <p>タスク遂行の責任： アーキテクトとして活動するテックリード</p>
反復のプランニング	<p>すべての反復は、開発のためにすべてを明確化するためのプランニングから始まります。プランニングには、ユーザーの導入を促進するための最近のユーザーフィードバックを含む必要があります。</p> <p>タスク遂行の責任： プロダクトオーナー</p> <p>助言者： テックリード</p>

反復のデリバリー	<p>反復のプランニングの後には、バックログに従い、アジャイルのベストプラクティスに基づいて開発が実施されます。</p> <p>タスク遂行の責任: プロダクトオーナー</p> <p>助言者: テックリード</p>
反復のテスト	<p>反復のために開発された機能は、アプリケーションの品質を保証するためにテストする必要があります。このフェーズは、反復のデモを実行する前に必要です。</p> <p>タスク遂行の責任: テスターとして活動するテックリード/テスター</p> <p>助言者: テックリード</p>
本番へのロールアウト	<p>開発されるアプリケーションには、アプリケーションの展開方法と使用方法を定義する計画が必要です。これには、ロールアウトの成功を判断するための基準と、ロールバック計画を始動する条件を特定するための基準が含まれます。</p> <p>タスク遂行の責任: プロダクトオーナー</p> <p>助言者: テックリード</p>

デジタルエクスペリエンスのための調整

活動	概要
モバイルアプリケーションの着手	<p>最初の機能要件と非機能要件、大まかな機能、ターゲットデバイスをキャプチャーしたら、ユーザーストーリーを定義してバックログを作成します。</p> <p>タスク遂行の責任: プロダクトオーナー</p> <p>助言者: ビジネスユーザー</p>
モックアップ - 低忠実度/高忠実度	<p>MVP定義フェーズでは、最初に低品質のモックアップでアイデアを提示する必要があります。モックアップの一部は、プロジェクトのステークホルダーがアプリケーションのビジョンに連携できるように高品質にしていきます。</p> <p>タスク遂行の責任: カスタマージャーニーデザイナーとして活動するUX/UIデザイナー</p> <p>助言者: ビジネスユーザー</p>
カスタマージャーニー	<p>MVP定義フェーズでは、ジャーニー、マルチタッチポイントのマッピング、エクスペリエンスマップの価値、ビジネス価値の指標を定義する必要があります。</p> <p>タスク遂行の責任: カスタマージャーニーデザイナーとして活動するUX/UIデザイナー</p> <p>タスク完了の最終的説明責任: プロダクトオーナー</p> <p>助言者: ビジネスユーザー</p>
ルックアンドフィールの実装	<p>MVP定義フェーズでは、高品質のモックアップに基づいてルックアンドフィールを実装する必要があります。これにより、将来のアプリケーションの外観についてビジョンが与えられ、開発プロセスを高速化できます。</p> <p>タスク遂行の責任: フロントエンド開発者</p> <p>タスク完了の最終的説明責任: テックリード</p> <p>助言者: カスタマージャーニーマネージャーとして活動するUX/UIデザイナー</p>

ネイティブプラグイン開発	<p>プロジェクト実施中に、モバイルデバイスの機能を必要とする要素を実装しなければならないことがあります。その場合は、ネイティブ開発したプラグインをアプリケーションに追加する必要があります。</p> <p>タスク遂行の責任：フロントエンド開発者</p> <p>タスク完了の最終的説明責任：テックリード</p>
品質管理	<p>開発後は、コードとアーキテクチャーの品質を保証する必要があります。</p> <p>タスク遂行の責任：アーキテクトとして活動するテックリード</p> <p>助言者：開発者</p>
機能テスト	<p>開発後は、機能テストのフェーズで、開発された機能が要件に沿っていることを保証します。</p> <p>タスク遂行の責任：テスター</p> <p>タスク完了の最終的説明責任：プロダクトオーナー</p> <p>助言者：テックリード</p>
デバイステスト	<p>開発後は、デバイステストのフェーズで、開発された機能が必要なモバイルデバイスに正しく実装されていることを保証します。</p> <p>タスク遂行の責任：テスター</p> <p>タスク完了の最終的説明責任：プロダクトオーナー</p> <p>助言者：テックリード</p>
負荷テスト	<p>開発後は、負荷テストのフェーズで、開発された機能が複数ユーザーのアクセスをサポートできるように正しく実装されていることを保証します。</p> <p>タスク遂行の責任：テスター</p> <p>タスク完了の最終的説明責任：DevOpsエンジニアとして活動するテックリード</p>
フィールドテスト	<p>開発後は、フィールドテストのフェーズで、開発された機能が正しく実装され、ユーザーが期待するおりにフィールドで動作することを保証します。</p> <p>タスク遂行の責任：テスター</p> <p>タスク完了の最終的説明責任：プロダクトオーナー</p>

稼働

活動	概要
本番でのリリース	<p>アプリケーションは本番環境に展開する必要があります。スタンドアロンのアプリケーションでない場合は、他の本番システムまたはアプリケーション構成に接続する必要があります。このプロセスは、リリースのロールアウトドキュメントに基づいて実施され、展開プロセスの確実な成功と時間短縮を実現します。</p> <p>タスク遂行の責任：DevOpsエンジニアとして活動するテックリード</p> <p>タスク完了の最終的説明責任：プロダクトオーナー</p>
アプリケーションのプロモーション	<p>アプリケーションの成功とユーザーの導入を確実に実現するには、ユーザーやステークホルダーに対してアプリケーションの社内的なプロモーションを行うことが非常に重要です。ビデオ、アプリケーションジャーニーのデモなど、必要なすべての社内PR活動により、社内マーケティングキャンペーンを作成し、最適なアプリケーション/ソリューションであることを（役員、IT幹部などに）示します。</p> <p>タスク遂行の責任：導入推進者</p> <p>タスク完了の最終的説明責任：プロダクトオーナー</p>

アプリケーションのフィードバックの収集	<p>アプリケーションを本番でリリースした後は、将来のリリースでの改善方法を特定したり緊急の問題を修正したりできるように、ユーザーの意見を監視する必要があります。機能、ユーザーエクスペリエンス、アプリケーションの問題などに関連するフィードバックを収集します。</p> <p>タスク遂行の責任：プロダクトオーナー</p> <p>タスク完了の最終的説明責任：テックリード</p>
アプリケーションのトラブルシューティング	<p>アプリケーションに関する問題が検出された場合は、症状の分析を実行し、根本原因を特定し、問題を緩和できなければなりません。問題に対処できない場合、アプリケーションは導入されず、成功することもありません。</p> <p>タスク遂行の責任：DevOpsエンジニアとして活動するテックリード</p> <p>タスク完了の最終的説明責任：プロダクトオーナー</p>
ビジネスケースの指標の評価	<p>本番稼働中のアプリケーションを要求フェーズで定義されたビジネスケースと指標と比較して、アプリケーションの成功度を判断します。導入率を判断し、その所見に基づいて計画を策定します。</p> <p>タスク遂行の責任：プロダクトオーナー</p> <p>助言者：導入エキスパート</p>
アプリケーションのトラブルシューティング	<p>将来的に、アプリケーションを進化させたり、新しいリリースをリリースしたり、新しい機能を提供したりする必要が生じる可能性があります。ポートフォリオを定期的にレビューすることで、次のステップを決定して優先付けできます。</p> <p>タスク遂行の責任：プロダクトオーナー</p> <p>助言者：導入エキスパート</p>

デジタルエクスペリエンスのための調整

活動	概要
ストアでのリリース	<p>必要に応じて、対応するストアでモバイルアプリケーションのプロモーションを行う必要があります。</p> <p>タスク遂行の責任：プロダクトオーナー</p> <p>助言者：ビジネスユーザー</p>
パフォーマンスの監視	<p>将来的な制約となる可能性のある要素を特定し、発生する可能性のある問題に（報告される前であっても）対処するために、アプリケーションを継続的に監視する必要があります</p> <p>タスク遂行の責任：DevOpsエンジニアとして活動するテックリード</p> <p>助言者：プロダクトオーナー</p>
ストアでのレビューの監視	<p>アプリケーションをストアでリリースした後は、将来のリリースでの改善方法を特定したり緊急の問題を修正したりできるように、レビューを監視する必要があります。</p> <p>タスク遂行の責任：プロダクトオーナー</p> <p>助言者：ビジネスユーザー</p>
導入の監視と調整	<p>モバイルアプリケーションが本番稼働を開始したら、ユーザーの導入状況を測定し、導入を促進するために何をすべきかを特定します。</p> <p>タスク遂行の責任：プロダクトオーナー</p> <p>助言者：ビジネスユーザー</p>



www.outsystems.com

© Copyright OutSystems 2018. All rights reserved. OutSystemsおよびOutSystemsのロゴは、OutSystemsの登録商標です。他のすべての商標は、それぞれの企業の所有物です。許可されない複製または配布は著作権法違反となります。

