

クラウド時代の新しいアプリケーション開発 イノベーションを創造する「IBM Cloud Garage」

「IBM Cloud Garage」はクラウド上に短期間でMVP(Minimum Viable Product)と呼ばれるプロトタイプを開発するための支援サービスで、その目的は顧客体験に基づく新しいビジネスを創出することです。デザイン・シンキングの手法を用いてアイデアを創造し、アイデアの実証に必要な最低限の機能を実装したMVPでエンドユーザーに価値を提供することができて、初めて成功とみなされます。IBM Cloud Garageでアプリケーションを開発する際は、自由な発想で開発する必要がある一方で、実サービス展開においては考慮しなければならないルールがあります。

本稿では、IBM Cloud Garageの本質および、実サービス展開でのポイントを解説します。

▶▶ 1. デザイン・シンキングとは

アプリケーションやサービスを開発する際の手法の一つである「デザイン・シンキング」という言葉を耳にしたことがあるでしょうか？ ここでの“デザイン”は、図案など工業意匠としてのデザインというよりも“設計”という意味合いが強く、“ビジネス上の問題を解決するための設計”と考えるとよいでしょう。

IBMではお客様のビジネス上の問題を解決するためにデザイン・シンキングを取り入れています。IBM内のファシリテート・チームやお客様を担当するIBMメンバーだけでなく、お客様の担当者や役員、場合によってはお客様のお客様(=エンドユーザー)も参画し、ユーザーのペルソナ

を定義した上で、ユーザーの困っている点を理解し、その問題点をどのように解決するかをデザインしていきます。

このデザイン・シンキングのプロセスにおいては、徹底して“ユーザー”目線を意識します。特定の手法や方法論を用いるのではなく、あくまでユーザーに向き合っており、ユーザーが困っている問題点を解決することがゴールとなります。デザイン・シンキングの参加者は全員がこの同じゴールを目指してアイデアの拡散・集中を繰り返します。そして生まれたアイデアの卵に対して、実際にサービス化する上でのユーザー体験やワイヤーフレーム、アプリケーションの使いやすいインターフェースなどを紙芝居のような感覚で作り上げて、後述のMVP実装チームと連携していきます。



図1. 駐車場問題に対して、IoTテクノロジーだけでは解決できなかった例

▶▶ 2. アイデアの創造:デザイン・シンキングの例

デザイン・シンキングをより正しく理解するために、実際に行われたデザイン・シンキングの例を紹介します。与えられた課題は、「慢性的な駐車場不足が深刻な問題となっている米国サンフランシスコ中心部で、駐車場の問題を解決したい」というものです。

サンフランシスコは東京並みの人口集中地域であり、東京をはるかに上回る車社会です。またシリコンバレーに代表されるように人口増加地域でもあり、土地価格は上昇し、駐車場を作るための土地確保さえ難しくなっています。「ちょっと買い物」のつもりで車で中心部に出かけても、駐車場の確保が非常に難しいといったことが起きています。この問題をITで解決できないか、というのが課題です。みなさんなら、これを解決するためにどのようなアイデアが浮かぶでしょうか？

まず初めに、この問題にテクノロジー面から取り組み過ぎてしまったアイデアを紹介します。最近はやりのIoT技術を応用して、各駐車エリアにセンサーを設置し、駐車場の空き/埋まり具合をリアルタイムに把握し、それを地図上にマッピングするというものです(図1)。確

かにIoT技術を使うことで実現できそうですし、これが実現すると、どこに空き駐車場があるのか、埋まり具合はどの程度なのかをリアルタイムに把握できて、空き駐車場を見つけやすくなる気がします。

しかし、このアイデアには大きな問題があります。これは「IoTテクノロジーを応用するアイデア」であって、「ユーザーの問題を解決するアイデア」ではないということです。事実、このアイデアを実装したところで地図上に表示されるのは図1の結果のように満車を示す黒い丸で埋まった地図で、「本当に駐車場が足りない」ということが証明されただけの結果になってしまいました。

一方、この課題に対してアイデアを拡散していく中で、「車を小さく縮めて、ポケットの中に入れておく」というアイデアを出した人がいました。一見すると実現性を無視した、突拍子もないアイデアのように思えます。しかし、このアイデアは実現性こそ疑問に感じるものの、解決策としては成立しています。車をポケットに入れてしまえば、駐車場の心配をする必要もなく、安心して用件を済ませることができ、用件が済んだらポケットから車を出して元に戻して、また運転して帰ればいいのですから(図2)。

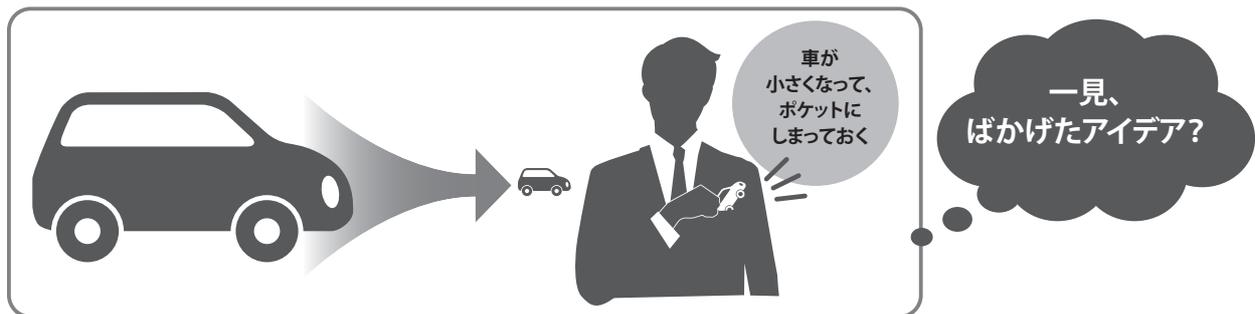


図2. 一見突拍子もないが、解決策として成立しているアイデアの例



図3. 突拍子もないアイデアからの気付きを実現可能なサービスに変える

このアイデアは、示唆に富んでいます。ユーザーは「駐車場に車を止めたい」わけではないということです。あくまでも自分の用件をしている間に、車が壊されたり駐車違反になったりせず安全な状態で確保されていればよいのであって、駐車場に停めておくのはその実現手段の一つでしかありません。駐車場以外に別の方法があればユーザーは満足するかもしれない、ということです。

ここに気付けばアイデアが広がるのに時間はかからないかもしれません。実際この時には、車を止めなくなったユーザーのところに、別の運転手がオンデマンドで車を取りに行き、約束した時間と場所に再度車を届けるというサービスのアイデアが生まれました(図3)。このサービスはどちらかというとITというよりも人海戦術を駆使する方法ですが、ユーザーの目的を達成するだけでなく、車を届けるまでの間は自由にその車を使えるため、車を所有していない運転手にとってもメリットがあるということになりました。

▶▶ 3. アイデアの検証:MVP

デザイン・シンキングで生まれたアイデアは本当に役立つものなのでしょうか？ あるいは本当に実現可能なものなのでしょうか？ 本格的に実装する前にアイデアの実証に必要な最低限の機能を盛り込んだ製品(Minimum Viable Product : MVP)を作ります。この段階においてはスケーラビリティやセキュリティと

いった非機能要件を検討するよりは、「動くものを作って試す」ことを軸として作ります。場合によってはフィードバックを基に改良を加えたり、思っていたような成果が得られなかった場合は傷の浅い段階でその機能をあきらめるといった判断を行うための製品化です。

前述のデザイン・シンキングでの例では、オンデマンドで車を受け取って返却の契約を行うシステムのMVPを作ることになります。具体的にはユーザーがスマートフォンを使うことを想定して、予約や車の確認を行えるシステムを実装します。加えて運転手の管理や、位置情報から地図へのプロット、モバイル化といった機能に優先順位を付けながら実装していくことが、この例でのMVP開発となります(図4)。

▶▶ 4. IBM Cloud GarageとCloud Innovateとは

IBM Cloud Garage(以下、Garage)は、デザイン・シンキングの手法を用いて、顧客体験に基づく新しいビジネスを創出することを目的としたクラウドのサービスで、前述のMVPの定義とプロトタイプの開発を範囲としています。

MVPを定義する際は、「ユーザー・プロフィール」や、ユーザーの決定に至るまでの行動や心理を時系列的に可視化した「ジャーニー・マップ」、ユーザーの発言、行動、思考、感情の4つの項目から成る「共感マップ」などを定

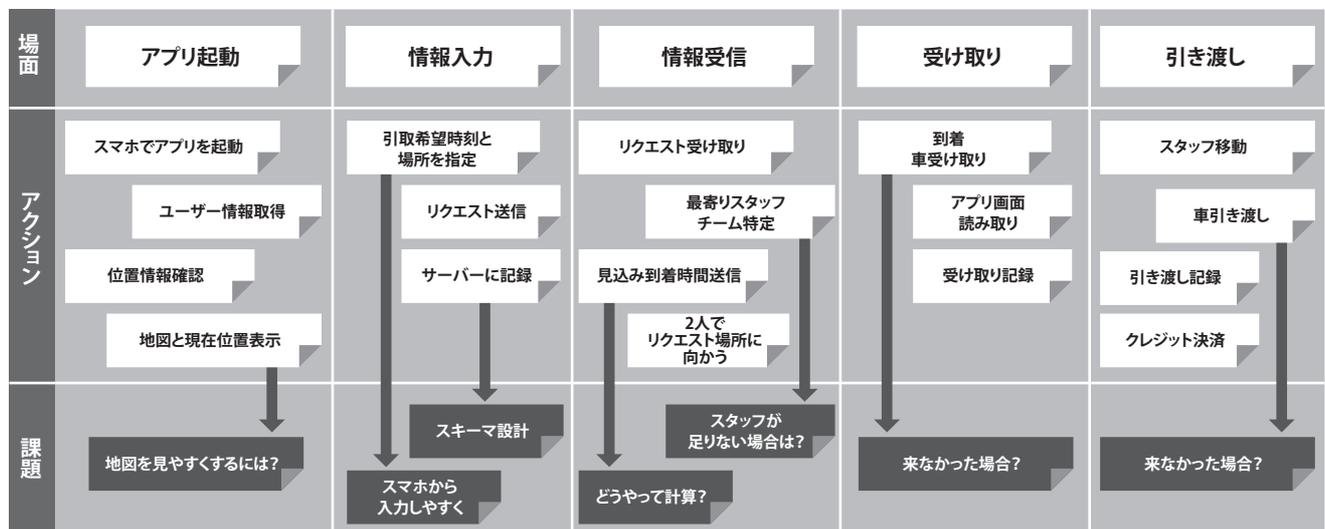


図4. MVP開発で実装するストーリーから個別のアクションと課題を切り出していく

義し、顧客体験を徹底して追求したプロトタイプを開発します。しかし、イノベティブなプロトタイプが開発されたとしても、実際にサービスとして展開できなければ絵に描いた餅です。デザイン・シンキングを実施している時は自由な意見を言い合い、お祭りのように楽しい時間が過ぎますが、MVPの目的を見失い実サービスやビジネスとして成り立たないケースも少なくありません。そのため、実際に開発されたプロトタイプからサービス展開するためには別の手法が必要です。IBMでは、クラウドに関わるビジネス・デザインやシステム・デザインから運用までを、6つのプロファイルから成り立つ「Cloud Innovate」(表1)(図5)として提唱しています。

GarageはCloud Innovateが提唱されるよりも前から存在していますが、Cloud InnovateではGarageは「Build on cloud」プロファイルのベースとなっており(図6)、前述のとおりデザイン・シンキングで定義したアイデアをMVP開発の中で実現可能かを検証します。

デザイン・シンキングとMVP開発を実行する前後で、特にMVP開発で作成したコードを再利用する場合は、簡易的な合意でもよいので「Advice on cloud」プロファイルで以下の項目について確認しておく必要があります。

- 法令・ガイドライン
- セキュリティー
- アーキテクチャー
- ROI

表1. Cloud Innovateのプロファイル

プロファイル	概要
Advice on cloud	クラウド適用に関するビジネス戦略策定
Migrate to cloud	アプリケーションを変更せずにインフラのみクラウドへ移行
Modernize for cloud	最新技術を取り入れクラウドネイティブなアプリケーションとして再構成
Rationalize for cloud	クラウド環境で複数アプリケーションを統廃合
Build on cloud	クラウドネイティブなアプリケーションとして新規開発
AppOps on cloud	クラウド環境での運用

SQuaRE (Systems and software Quality Requirements and Evaluation)や非機能要求グレードの項目のすべてを実サービス開始時点からすべて備えることは難しく、備えた場合はビジネスとして立ち上げ時から黒字化することは不可能でしょう。しかし、特に「法令・ガイドライン」とデータ漏洩対策が主軸の「セキュリティー」の対応についてはビジネスとして順守が必須のため、ビジネス責任者とMVP開発チームで合意する必要があります。なお、MVP定義したアプリを開発しリリースする際に一から作り直すということができるのであれば、最初から順守する必要はありません(図7)。

次章以降では、各項目について説明します。

5. 法令・ガイドライン

クラウド・ベンダーはクラウド上に特定のデータを保管することを禁止しておらず、データの取り扱いがクラウド利用者が責任を持つため、クラウド利用者は法令とガイドラインについて対応の是非を求められます。代表的な法令としては、国内では個人情報保護法、海外ではGDPR(General Data Protection Regulation)などが挙げられます。MVPを実装してもサービス展開できなければ意味がないため、MVPの定義後に法令とガイドラインを順守可能かを早期に見極める必要があります。

代表的な公的なガイドラインとしては、国内では「FISC(The Center for Financial Industry Information

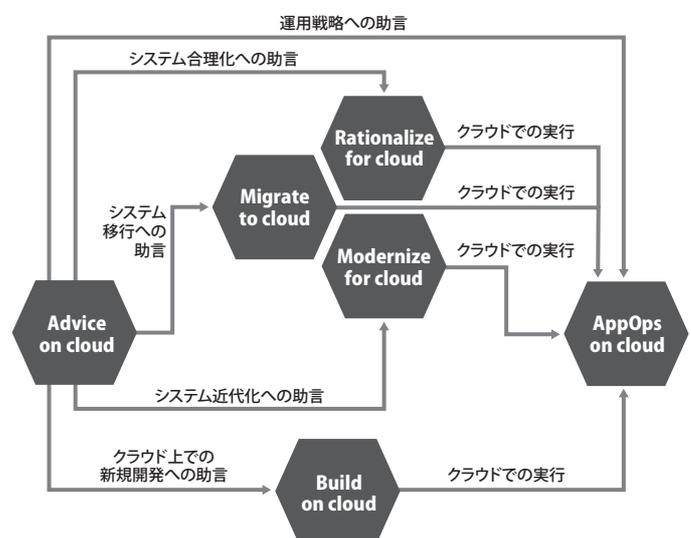


図5. Cloud Innovateのプロファイルの関連

Systems)安全対策基準」や「3省4ガイドライン」、海外では「PCI DSS (Payment Card Industry Data Security Standard)」が挙げられます(技術解説「IBM Cloudのレギュレーション対応」64ページ参照)。

MVP開発においてこのガイドラインが課題となることが多くあります。なぜならば、ガイドラインはシステムの詳細に踏み込んで順守すべき項目を定義していることが多いからです。ガイドラインは法令上に定められた罰則がないためMVP開発時点では後回しにされ、実サービス展開時に企業のサービスの信頼が損なわれることを防ぐため高いコストを支払って対応することになります。

Garageは斬新なアイデアを尊重するため、MVPの定義段階で法令とガイドラインを前提とする必要性はありません。しかし、MVPのシステム・アーキテクチャーを決定する初期スプリントでガイドラインを順守すべきか否かを見定め、必要であれば本番化に向けてのリファクタリングやクラウドの基盤において、対応を取り込む必要があります。

例えば、医療情報に関するガイドラインである3章4ガイドラインでは、本執筆時点でIBM CloudのIaaSが対応していますがPaaSは対応していません。PaaSの場合はクラウドとしての監査ログの管理などの機能がマルチ・テナント化や利便性のため隠蔽されており、対監査性が強く求められる業界では適用が難しい場合があります。MVPが確定した時点でそのアイデアが上記のような法令を順守し

つつ、ビジネス展開できるかを検証する必要があります。

6. セキュリティー

独立行政法人情報処理推進機構が公開している非機能要求グレード(可用性、性能・拡張性、運用・保守性、移行性、セキュリティ、システム環境・エコロジー)のうち、優先されるべきはセキュリティです。なお、他の項目の優先順位はビジネス責任者との協議により決定します。例えば、新しいサービスでアプリケーション・エラーやアクセス不能を経験された方も多いと思いますが、これは顧客体験を重視し、品質を含む可用性が後回しになっているためです。しかし、セキュリティについては、セキュリティ事故などでビジネスのブランドに傷がつく前に議論されるべき項目です。セキュリティの検討項目は多岐にわたりますが、インフラにおける個々の脆弱性や攻撃に対する対策としては、WAF(Web Application Firewall)、IPS(Intrusion Prevention System)/IDS(Intrusion Detection System)、SIEM(Security Information and Event Management)などが挙げられます。

前述したように、MVPを基に開発したコードに関して実サービス展開時にアプリケーションを一から作り直せるのであれば、セキュリティを最初から検討する必要はありません。しかし、新ビジネスは競合他社が同様のビジネスを始める前に急速に展開する必要があります。そのためコードを作り直している時間はなく、リファクタ

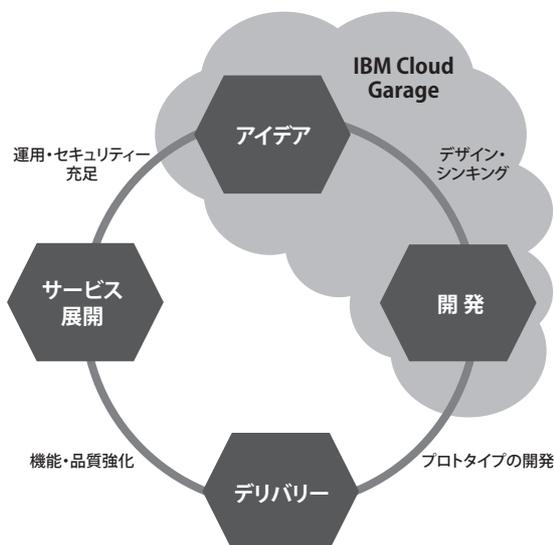


図6. Build on cloudの活動とIBM Cloud Garageの関係

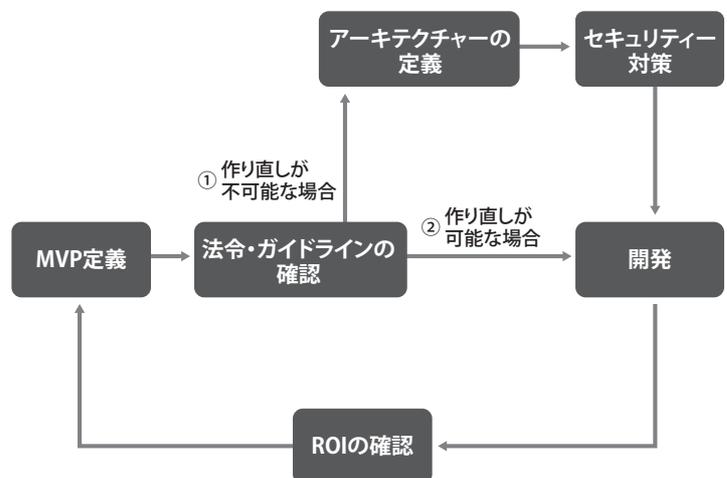


図7. MVP定義から開発に移るまで

リングを行いながら常に作り変えていくこととなります。そう考えると、アプリケーションとしてのコードが複雑化し手直しできなくなる前に、MVPが確定した時点でセキュリティをどのレイヤーで担保するのかを確認する必要があります。

▶▶ 7. アーキテクチャー

MVP開発を行う場合は、初期スプリントの中で当面のアプリケーション・アーキテクチャーを決定します。作り直しが許容されないのであれば、後々のリファクタリングや機能拡張が容易となるアーキテクチャーを採用すべきです。典型的な例としてはAPI化やマイクロサービス化などのアプリケーション・アーキテクチャー(技術解説「マイクロサービス構築を始めるには」48ページ参照)、コンテナ技術やInfrastructure as codeなどのインフラ・アーキテクチャー(技術解説「コンテナ・オーケストレーションでシステム・モダナイゼーションを加速する!」30ページ参照)が挙げられます。

筆者の経験としてアーキテクチャーを決定しないことで将来問題となることが多いのが、アプリケーション・レベルでの権限管理です。インフラ・レベルや運用での権限管理は6章でも述べている非機能要件の考察の際に定義されるはずですが、MVPを開発する際において、機能と直接関係のないアプリケーション・レベルでの権限管理は優先順位が低くなりがちです。一方、斬新なサービスは個人情報など重要なデータを扱いユーザーへの表示/非表示の要求は後々高まるが多くなります。この際、表示/非表示処理を画面のHTMLやビジネス・ロジックに個別に埋め込むと実際のサービス開始前に大幅なリファクタリングやテストケースの増大の原因となります。サービスを利用するアクターをMVP定義時にすべて見通すことは困難ですが、初期のアーキテクチャー策定には最新の注意を払い、常にアーキテクチャーの刷新があることを意識する必要があります。

▶▶ 8. ROI

MVP定義時点でROI(Return On Investment)を正確に算出することはほぼ不可能です。革新的なサービスは、思いつきから発生するものであり思いつきをいかに

早期に世界に発信するかが問われます。しかし、実際には、「まずはやってみる」という社の号令のもとで開始したとしても、いざMVPができ上がるとROIが成立しないということで中断することが多々あります。また、世のスタートアップも実は赤字で外部からの投資で運営資金を賄っているというケースは多くあります。新しいビジネスは全体の数%が成功と言う世界であり、ある意味で賭けです。成功率を高めるためのMVP開発ですが、ビジネス責任者はデザイン・シンキングの実施時から常にROIを意識しておくことが重要です。

▶▶ 9. おわりに

ここまで、デザイン・シンキングでは徹底したユーザー目線でのアイデアの創出が重要であること、実サービスの展開では後々大きな問題となりがちなポイントについて例を交えながら、Garageをベースとした新しい開発手法を紹介してきました。

誤解されがちなのは、常にこうした新しい開発手法が必要なわけではないということです。ビジネス要件に議論の余地がない場合は、ウォーターフォールによる要件定義から設計・開発を行えば良いのです。Garageはグローバル化・技術革新など将来が予測できない社会の中で、顧客体験を重視したビジネスを展開するときに有効な開発手法であるということです。



日本アイ・ビー・エム株式会社
グローバル・ビジネス・サービス事業部
クラウドネイティブ開発
アドバイザー・アーキテクト

石井 真一
Shinichi Ishii

日本IBM入社後、オープン系システム構築やソフトウェア開発を経てクラウドでのアプリケーション開発とインフラ構築に従事。常に新しいことにチャレンジし挫折を繰り返しながら日々精進中。



日本アイ・ビー・エム株式会社
研究開発
IBMクラウドガレージ
ガレージ・デベロッパー

木村 桂
Kei Kimura

IBM Cloud エバンジェリスト兼ガレージ・デベロッパー。マンホールを愛し、マンホールに愛されたマンホールラーでもあり、(自称)世界最大の位置情報付きマンホール情報共有サイト「マンホールマップ」を開発・運営中。