

趣味演奏者向けSNS「music演人」の開発

Development of Social Networking Service “The MusicEngine” for hobby musicians

宮田 明, 大久保 英幸, 清水 晃, 城崎 康夫, 新宅 貴行, 福島 章雄
Akira Miyata, Ohkubo Hideyuki, Akira Shimizu, Yasuo Shiroasaki, Takayuki Shintaku, Akio Fukushima

要旨 趣味として演奏を楽しむ方々に向けたSNS「music演人」を開発した。music演人の参加者は演奏を録音・共有し、擬似的な合奏やコミュニケーションを行う。music演人はサーバー・クライアント型のシステム構成を取り、クライアントはサーバーで公開するAPIとウェブコンテンツを用いてWebアプリケーションを構築する。およそ1年の公開実験の結果、イベントの継続的な推進と、録音の難しさを解決しなければ、コミュニケーションの活性化は期待できないことが分かった。また、新規のモノ・サービスを検討する場合、ウォーターフォール型の開発プロセスは不適であることが分かった。

Summary We have developed Social Networking Service “The MusicEngine” for people who perform music for fun. Users on “The MusicEngine” can record and share their own musical performances and have communications among users. From a system point of view, “The MusicEngine” consists of client-server architecture. Clients access a server open API, get web contents, and display the contents to users. Through the one year open experiment of “The MusicEngine”, we have learned the followings.

1. The procedure for sound recording must be easy. .
2. Attractive events are important to gather users continuously in the early stages of a service. Without achievement of above 2 items, self-motivated communications among users never happens. In addition, we have also learned that Waterfall development process does not suit for new services or products.

キーワード Webサービス, SNS, クラウド, 開発プロセス

1. はじめに

本開発は趣味領域における新規ビジネスの創造を目的としたものである。我々は、数ある趣味の中からパイオニアの保有する音楽解析技術と親和性が高い「音楽演奏」の領域に焦点を当て、演奏の上達や新たな演奏仲間との出会いをサポートする機材やサービスの提供を目指し開発を行った。本稿では、公開実験を行った趣味演奏者向けSNS「music演人」と、そのサービス運用中の取り組みについて報告する。また、本サービスは継続的な実施には至らなかったが、その結果に対する考察を述べる。

2. music演人の概要

2.1 music演人の目的

music演人は趣味として演奏を楽しむ方々に向けたソーシャル・ネットワーキング・サービス(SNS)である。「演奏をする機会がない」「演奏仲間と出たい」といったニーズを満たすことがサービスの目的である(図1)。ユーザーはmusic演人で楽曲を演奏したり、演奏を録音して共有する

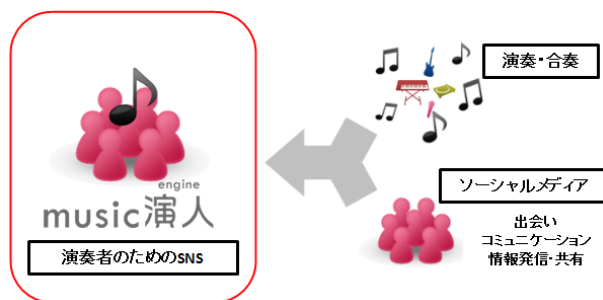


図1 music演人の概念



図2 music演人のホーム画面

ことで、擬似的な合奏やコミュニケーションを行う。図2は music演人のホーム画面のスクリーンショットである。

2.2 music演人の基本概念

図3はmusic演人で扱う演奏データを表した概念図である。パート(Part)は楽器ごとの録音データ、バンド(Band)は複数のパートで構成されるグループ、セッション(Session)は演奏におけるルールを表す。ここで述べたルールとは、例えば曲目、キー、テンポ、曲の時間的な長さ等が挙げられる。これらのルールを遵守して各楽器の演奏を録音しパートとして共有することで、同じセッション内の複数のパートを違和感なくミックスすることができる。例えば、あるセッションにドラム、ベース、ピアノのパートが複数個ずつ共有されている場合、それぞれの演奏パートから好きなパートを選んでバンドを構成し、そのバンドを伴奏として演奏を楽しむことができる。この基本概念を元に、music演人の機能やユーザーインターフェースを検討し開発を進めた。

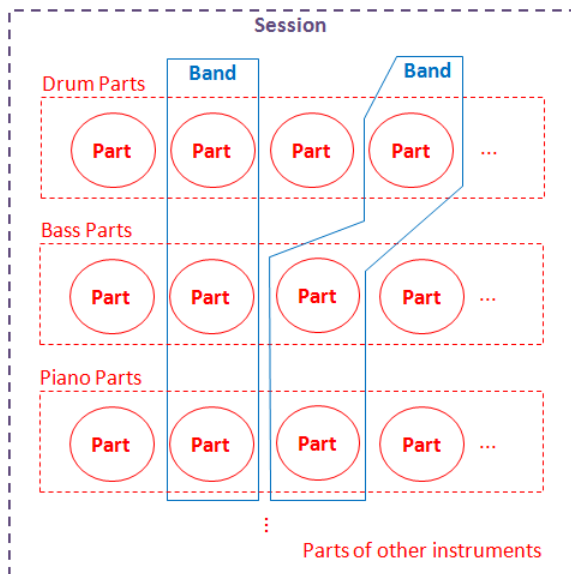


図3 演奏データの概念図

2.3 music演人の構成

music演人はインターネットを通じたサーバー・クライアント型のシステム(図4)であり、クライアントにはWebブラウザを利用する。演奏の録音はオーディオ・インターフェースをPCに接続し、Webブラウザ上で動作するFlash*アプリケーションを用いて行う。詳しいシステム構成に関しては2, 3章で述べる。

2.4 music演人の機能と使い方

music演人を利用する場合、ユーザーは最初に利用登録を行う。登録の際に、システムで用いる名前や演奏する楽器などのプロフィール情報を設定する。システム側が必

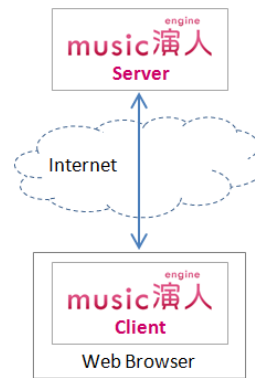


図4 システム構成

要以上に個人情報を持たないようにするために、ユーザーの認証にはOpenIDを採用した。ユーザーは登録後にmusic演人にログインし、演奏したいセッション(曲)と伴奏に用いるバンドを選択する。演奏の準備が整ったらバンドを再生し、バンドの演奏に合わせて演奏を楽しむ。music演人はJASRACから楽曲利用の許諾を得ており、JASRACの管理する曲、歌詞、楽譜であれば利用することができる。また、自作の曲も同様に利用することができる。演奏を共有したい場合には自身の演奏を録音し、録音データをmusic演人にアップロードする。この時、演奏に用いたセッションやバンドの情報も一緒に送信され、システムに登録される(図5)。尚、アップロードした演奏の利用、配信等の権利はmusic演人がユーザーから利用許可を得る形式とし、利用登録時に同意を頂くこととした。ユーザーは演奏の共有の他に、演奏に対するCool!(所謂「いいね!」)やコメントの付加、フレンド・お気に入りユーザー登録、特定ユーザーへのメッセージの送信等の機能が利用できる。music演人でユーザー同士のコミュニケーションを活性化させるために、これらの機能を開発した(図6)。

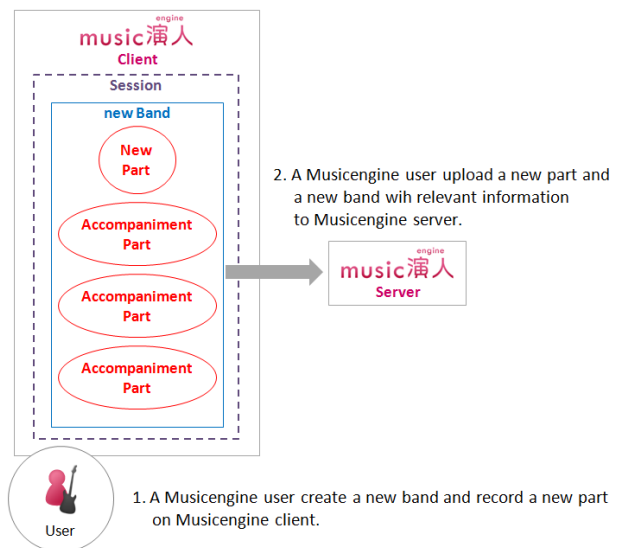


図5 演奏のアップロード

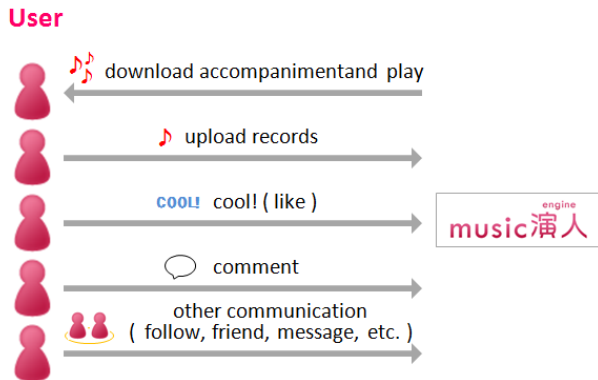


図6 music演人で可能なアクション

以下にmusic演人に搭載した主な機能を示す。(サービス公開後のアップデート時に追加した機能を含む。)

演奏に関する機能

- 演奏のマルチトラック再生
- ミキシング(パン、音量調整)
- 音量指標値を用いた自動音量調整
- 演奏の録音
- 録音した演奏の共有(パートの作成)
- お気に入りセッションの登録
- 新たなセッションの作成
- キー・テンポチェンジ再生
- 伴奏に連動した楽譜の表示(一部の楽曲のみ)

ユーザーコミュニケーションに関する機能

- プロフィールの作成、表示
- ユーザーのフォロー・フレンド登録
- 特定ユーザーへのメッセージの送信
- フォロー中・フレンドのユーザーやお気に入りセッションのアクティビティ表示
- 演奏に対するコメントの付加
- コメントのTwitter^{*}への投稿
- 演奏に対するCool!(いいね!)の付加

3. システム構成

3.1 サーバー側のシステム構成

図7にサーバー側システム構成のブロック図を示す。サーバー側の処理は主にPHPを用いて実装されており、Webサーバーで公開するAPIをhttpでコールすることで処理が開始される。APIには認証用API、各種データのgetter・setter用APIがあり、クライアントはこれらを必要なタイミングで呼び出す。データの形式にはXMLを採用し、クライアントから要求されたデータはサーバーから出力される際にパーサによってXMLに成型される。

クライアントから録音データがアップロードされると、録音データに対し各種の処理が行われる。まず、wav形式でアップロードされた録音データはmp3へ変換される。これは、

音声ファイルを圧縮することで扱うデータサイズを軽減し、アクセス速度を向上させるためである。次に、アップロードされた録音データを用いて試聴用のミックスダウンデータの作成、自動音量調整に用いる音量指標値の計算、UI向けの音声波形画像の作成が行われる。最後に、処理した各種データをRDBMSに登録して一連の処理が終了する。これらの処理を終えた後に、アップロードした録音データはパートとしてmusic演人で利用できるようになる。

プロフィール用画像がクライアントからWebサーバーにアップロードされると、画像を適切な画像形式・サイズに変更する。これは、音声ファイルと同様に扱うデータサイズを必要最小限にするためである。

認証用APIはユーザーがmusic演人にログインする際にクライアントからコールされる。APIがコールされた後、サーバーはOpenIDプロバイダにアクセスして認証処理を行う。OpenIDを採用することで、メールアドレス等の個人情報をサービス提供側で保持することなく認証機能を構築することができる。

Twitterへの書き込みは、サーバーがTwitterAPIを用いて行う。TwitterAPIの利用にはTwitterへのOAuth認証が必要であり、ユーザーは予めOAuthの利用を許可し、アクセス用のトークンをmusic演人に登録しておく必要がある。書き込んだ内容(Tweet^{*})は、書き込み時に得られるTweetIDが分かればTwitterAPIを用いずに取得できる。このため、TweetIDはRDBMSに保持するようにした。音声データや楽譜データにはワンタイムパスワードを用いてアクセスする。これはJASRACから利用許可を得ている楽曲が不正利用されないようにするためである。また、setter用APIにもワンタイムパスワードを利用し、不正なアクセスがされないようにしている。

クライアントで用いるJavaScript^{*}、Flashアプリケーション、HTML・CSS・画像データ等の静的ファイルはWebサーバーにて配信する。クライアントはこれらをWebブラウザを用いてダウンロードし、Webブラウザ上にWebアプリケーションを構築する。

サーバー側のシステムは容易にスケーリングが可能なAmazon Web Service^{*}のEC2^{*}に構築し、サーバー負荷の変化に対応できるようにした。

3.2 クライアント側のシステム構成

図8にクライアント側システム構成のブロック図を示す。クライアントとなるWebアプリケーションはGoogle Chrome^{*}ブラウザでのみ動作を保証した。数あるWebブラウザの中でGoogle Chromeブラウザのみに限定したのは、各種ブラウザに依存した作りこみを回避し、開発工数を抑えるためである。価値検証の段階ではクオリティよりもスピードが重要と考え、このような判断となった。また、他のブラウザに比べて、HTML5やデバッグといった先進的な機能が充実していたことも理由である。

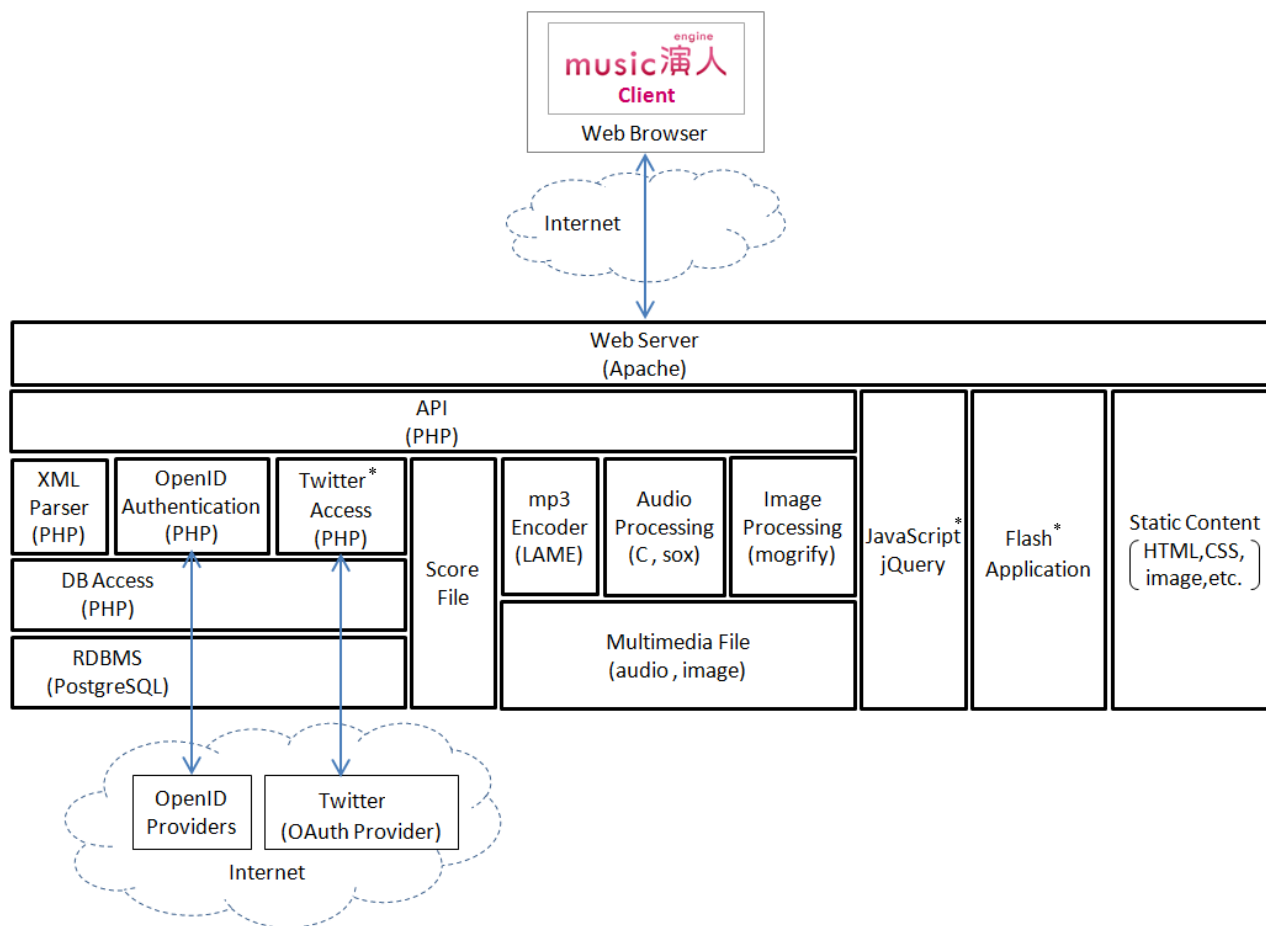


図7 サーバー側の構成

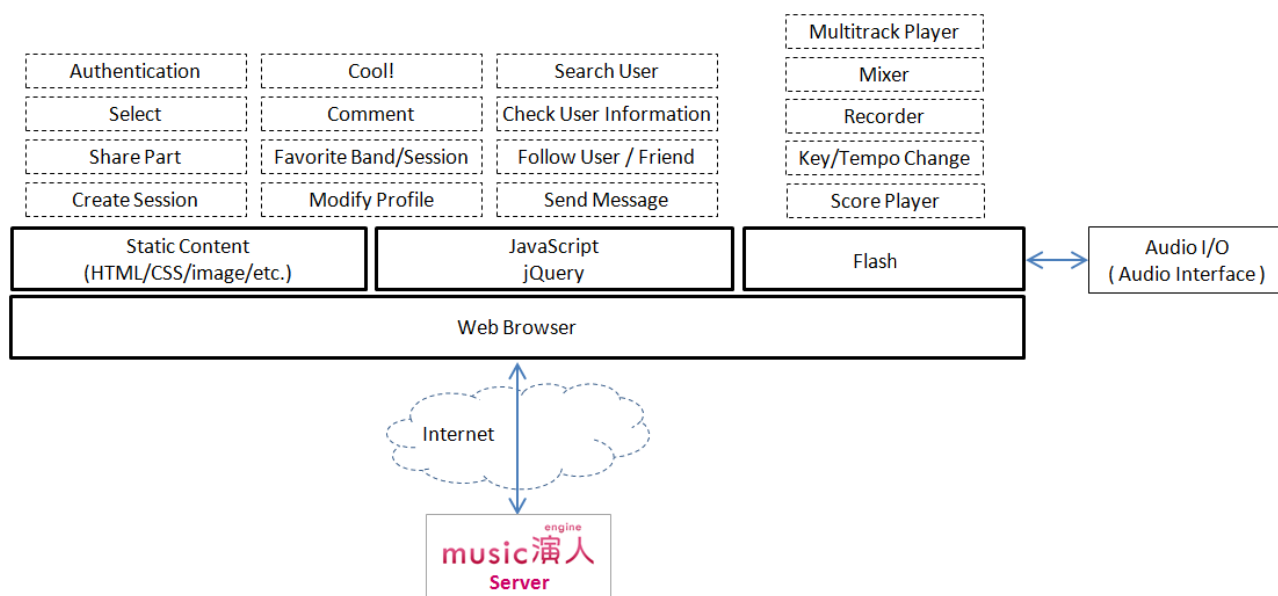


図8 クライアント側の構成

クライアントの設計はMVCデザインパターンを参考に行い、仕様変更に対して柔軟な対応ができるようにした。クライアントのUIはHTML・CSS・画像データ等の静的データとJavaScriptで構築される。UIの基礎となる部分は静的データで構築され、動的に構築する必要のある部分にはJavaScriptを用いる。music演人を利用するためには、最初にこれらをサーバーからダウンロードし、クライアントの基礎となる部分を構築する必要がある。ユーザーや楽曲に関するデータは、クライアントからAjaxを用いてサーバーのgetter用APIにアクセスすることで取得する。クライアントはボタンのクリック等のユーザーからのアクションを受け取ると、Ajaxを用いてサーバーから必要なデータを取得し、UIの再構築や音声再生等の処理を行う。

マルチトラック再生、ミキシング、録音、キー・テンポチェンジといった音声の再生に関わる機能は専用のFlashアプリケーションで行う。再生する音声データはこのFlashアプリケーションから直接取得するようにしており、その際にはAjaxを用いてサーバーにアクセスしている。また、録音データのアップロードもFlashアプリケーションを用いて行う。

伴奏に連動した楽譜の表示は専用のFlashアプリケーションで行う。予め、楽譜に表示するカーソル位置と音声再生位置の関係をファイルに書き出しおき、その情報を元にして音声再生中に楽譜の演奏位置にカーソルを表示する。(図9)この機能は、楽譜が登録されている一部の楽曲のみ有効にした。

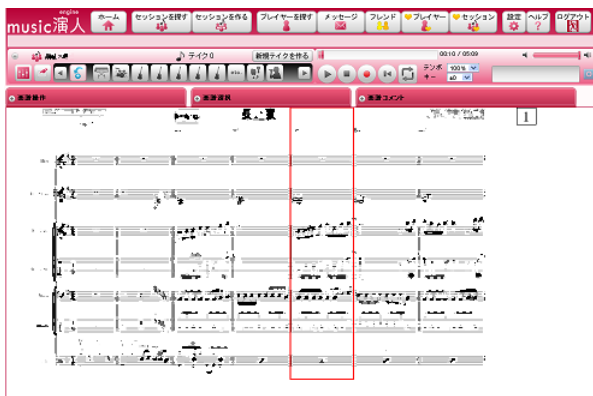


図9 楽譜の表示

4. music演人の公開と運用

4.1 当初の目的

2011年8月8日よりmusic演人の公開実験を開始した。公開当初の目的は、一般の趣味演奏者の方々にmusic演人を利用して頂き、演奏ソーシャルメディアとして機能するかを確認することであった。公開の時点ではテストが十分でなく、サービス品質が満足いくものではなかったが、ユーザーからのフィードバックを得ることが優先と考え、公開に踏み切った。公開開始直後でも直ぐに演奏が楽しめるように、music演人には予め100曲ほどのコンテンツを登録した。

4.2 サービス公開直後の状況と改善

図10に2011年8月から2012年3月までのmusic演人の登録者数とアップロード数、図11にmusic演人のユニークビジター数を示す。

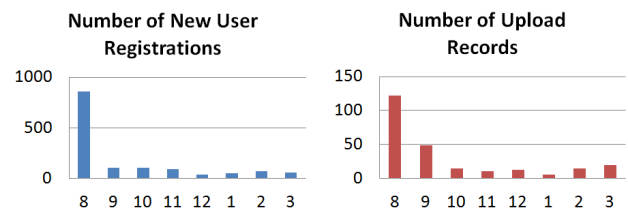


図10 登録者数と録音アップロード数の推移
(2011年8月～2012年3月)



図11 ユニークビジター数
(2011年8月～2012年3月)

公開直後である8月の登録者数は860人、アップロードされた演奏は122個に上った。しかし、9～12月にかけて登録者数、演奏のアップロード数、ユニークビジター数は減少傾向となった。これは、music演人のコンセプトの目新しさ、公開時に発表したプレスリリースの効果、そしてインターネット上のニュースサイト等で紹介されたことが要因で公開直後は好調であったが、時間の経過に伴いこの効果が薄れたためであると見られる。また、サービスの品質が低く、サービスを利用するユースケースも不明瞭であったため、music演人に参加したユーザーのコミュニケーションが進まなかったことも原因と見られる。

このため、12月より改善のための施策を行った。まず、サービス品質を高めるためにUIの大幅なアップデートを行った。アップデート前まではアイコンを中心にUIパーツを配置していたが、キャプションを付加して操作を分かりやすくした。そして、UIに関するモニターテストも実施し、基本的な操作に大きな問題点がないかを確認した。また、ユーザーがサービス利用へのモチベーションを持続させるために、新規セッションの定期投入や開発者ブログの公開など、サービスに常に動きを出すための施策を開始した。ここで投入した新規セッションは、これまで中心であったコピー演奏向けの楽曲ではなく、ジャズのようなジャムセッションで演奏される楽曲を中心とした。これは、ジャムセッションではアドリブが中心となり、コピー演奏に比べて自由度

が高いので、music演人のコンセプトに向いていると考えたためである。この結果、2012年1月から3月にかけて、登録者数、アップロードされた演奏の数、ユニークビジター数は増加傾向となった。

4.3 サービスの運用方針の修正

2012年3月までの結果を踏まえ、4月より運用方針の見直しを行った。3月以前はmusic演人の位置づけやターゲットユーザーについて不明瞭な点があり、ユーザーの利用促進のための具体的な施策が立て辛いという問題があった。このため、music演人の価値を「演奏に対するモチベーションを得られる場」として定義し、そのための仕組みを用意することやその阻害要因を除去することを運用方針とした。そして、music演人の利用に対するモチベーションを4種類に定義し(図12)、SNSを謳うmusic演人にとって特に重要な「後押しモチベーション」を喚起する仕組みを検討すべきとした。

種類	例
初期	目新しいコンセプトのSNSサイトに対する興味
目的	演奏したい曲があり、それを弾けるようになりたい
効果	伴奏や楽譜があるのですぐに練習が始められて効率が良い
後押し	誰かが演奏を重ねてくれた、またやってみよう

図12 モチベーションの分類

後押しモチベーションは他人との演奏によるコミュニケーション(足りないパートを補完、自分なりの演奏で置換、掛け合い、等)や、コメント等の演奏に対する評価によって喚起され、中でも特に重要なのは演奏によるコミュニケーションであると考えた。このため、自由度が高く、演奏によるコミュニケーションが発生し易そうなジャムセッション向け楽曲を以降のメインコンテンツとし、ジャムセッション愛好者を当面のターゲットユーザーとして施策を進めることにした。さらに、演奏によるコミュニケーションが発生するとユーザーに通知する機能(図13)や、ユーザーやお気に入りセッションに対するアクティビティ(コメント、Cool!, 演奏のアップロード等)を表示する機能(図14)を追加し、後押しモチベーションの向上を狙った。



図13 ユーザーへの通知機能



図14 アクティビティの表示

これまでの方針ではサービス基盤(インフラストラクチャー)と必要最低限のコンテンツを用意すればユーザーが集まり、演奏によるコミュニケーションを含めたmusic演人のエコシステムが自然と出来上がる、と考えていた。しかし、4月の時点ではエコシステムは出来上がっておらず、今後出来上がるとしてもこのままでは多大な時間が掛かると思われる。このため、エコシステム構築を促進する要素を用意する必要があると考え、以降の施策に盛り込むことにした。図15はこの施策を表した概念図である。

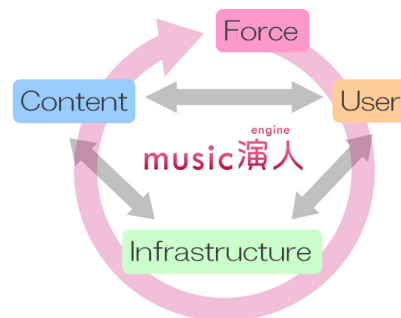


図15 music演人エコシステム構築のための要素

4.4 イベントの開催と結果

変更した方針に合わせて、ジャムセッションを学ぶ教室FTJS!を運営する株式会社ジャムコミュニケーション様に運営のコンサルティング、コンテンツ作成、テストユーザーの斡旋を目的として協力を依頼し、同社主導で8月1日から9月25日の期間、エコシステム構築を促進するためのイベントを開催した。ターゲットはFTJS!の生徒とし、イベント開始から終了までのアクティブユーザー数の推移を見た。イベントの内容は、演奏レクチャー用動画を動画共有サービスのYouTube*動画コミュニティにほぼ毎日アップし、動画でレクチャーした内容をmusic演人で演奏する、というものであった。

イベント期間内におけるアクティブユーザー数の2週間毎の推移を図16に示す。ここで、アクティブユーザーとは「二週間の間に『演奏をアップロードする』『演奏する(アップロードはしない)』『演奏を聴く』のどれかを行ったユーザー

一」と定義した。イベント期間の中頃にかけてアクティブユーザーは増加したが、イベントの収束にかけて減少傾向になった。

イベント期間内における機能別のユーザー数の推移(2週間毎)を図17に示す。試聴、再生を利用したユーザーはアクティブユーザー数と同様に推移しているのに対し、録音、アップロード、セッションの作成については大きな変化なく、低い割合で推移した。

イベント終了後に行ったアンケートからは、コンセプト面では大多数の回答者から共感を得られた。特に、練習への利用という点でポジティブな反応が得られた。一方で、録音に関して不慣れである点、オーディオ・インターフェースのセッティング等、録音の作業が難しい点を指摘された。

この結果から、イベント期間中において大多数のユーザーは「聞く」「演奏する」までに止まり、録音の難しさが原因で演奏をアップロードするまでに至らなかったことが推察された。最終的にアクティブユーザー数が増加したことから、イベントの効果があったことは証明できたものの、録音の難しさを解消しなければ演奏によるコミュニケーションの活発化は期待できないことが分かった。このため、イベント終了以降はmusic演人の運営は収束していくことになった。

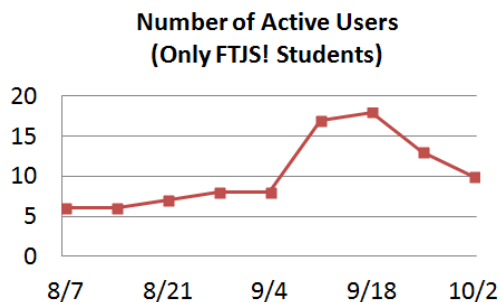


図16 イベント期間内におけるアクティブユーザー数の推移(FTJS!生徒のみ)

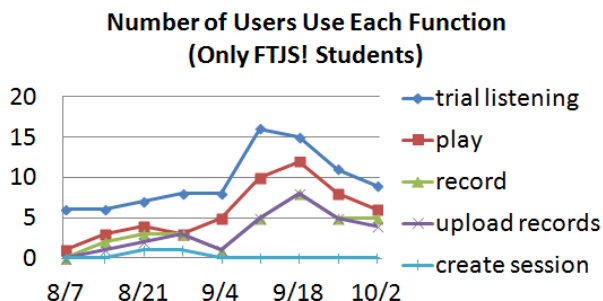


図17 イベント期間内における機能別のユーザー数の推移(FTJS!生徒のみ)

4.5 運用を終えての考察

music演人は2013年6月に公開実験を終了した。2012年10月以降も新規ユーザーの増加や利用が見られたが、エコシステムの定着は見られなかった。

今回、「演奏する機会がない」「演奏仲間と出会いたい」というニーズを満たすために、ウォーターフォール型のプロセスで、長い時間をかけてmusic演人を構築した。そして、1年ほどの運用を経て、music演人はこのニーズを満たせるものではないことがわかった。しかし、この結果はこれほどの期間をかけなくても得られたはずである。

例えば、リーン・スタートアップ⁽¹⁾では、必要最小限の機能を持つプロトタイプをユーザーに提供し評価することで、効率的に価値検証を行うように述べられている。そして、価値検証のプロセスを多く回すことでコンセプトや機能を何度も見直し、モノやサービスを構築していくべきと述べられている。本開発でも運用半ばからリーン・スタートアップを参考にし、本稿で述べた価値検証のための施策を始めた。しかし、検証にとって不要な機能がmusic演人には多いため、問題点がどこにあるのかが分かり辛く、改善のための修正も困難であった。つまり、高機能が仇となって工数が増大し、効率的な開発ができなかった。一方で、多大な工数を掛けて構築したmusic演人を簡単に捨てることもできず、半ば無理やりmusic演人を検証に用いる形となってしまった。例えば「演奏によるコミュニケーション」が成り立つかを検証するには、そのための機能に絞ったプロトタイプの方が適切であるし、開発工数も少なく済む。また、「演奏する機会がない」というニーズを満たすためには、music演人のようにインターネット上で演奏を共有するのではなく、ジャムセッションができる場所を教えるサービスの方がより適切だったと考える。このような検討を行っていれば、今回の結果がより早く得られ、music演人はまた違ったものになっていただろう。

5. おわりに

今回の検討から、新たなモノ・サービスを開発する場合には、ウォーターフォール型のプロセスは不適だということが分かった。多大な工数の掛かる開発を計画し行うのではなく、アジャイルなプロセスで検討を進め、徐々にサービスを構築していくべきである。また、ユーザーから正当な評価を得るために、利用に支障のない品質を維持することも必要である。

現在も演奏をサポートするモノ・サービスの検討は続けており、本開発で得られた知見や技術を元に即興演奏練習用のiOS*アプリ(図18)を検討・開発している。このiOSアプリは「Session Partners」という名称でApple社のApp Store*にて公開しており、「リーン・スタートアップ」の手法を参考にユーザーの反応を見ながら開発を進めている。本開発で得られた気付きや反省を元に、このiOSアプリを含め、今後新たなモノ・サービスを検討する場合には、ユー

ザーにとって価値があることを確認しながら開発を行っていく。



図18 即興演奏の練習用iOSアプリ

商標について

本文中の商品・サービス名は、各社の商標または登録商標です。

- * Twitter, Tweetは、Twitter, Inc. の米国およびその他の国における商標または登録商標です。
- * Flashは、Adobe Systems, Inc. の米国およびその他の国における商標または登録商標です。
- * Amazon Web Services, Amazon EC2は、Amazon.com, Inc. またはその関連会社の商標です。(※Amazon EC2は、Amazon Elastic Compute Cloudの略称です。)
- * JavaScriptは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標または商標です。
- * Google Chrome, YouTubeは、Google Inc.の商標または登録商標です。
- * iOSは、Apple Inc.のOS名称です。
- * IOS(iOS)は、Cisco の米国およびその他の国における商標または登録商標です。
- * App StoreはApple Inc.のサービスマークです。

参考文献

- (1) リース, E. 井口耕二訳 (2012) 『リーン・スタートアップ ムダのない起業プロセスでイノベーションを生み出す』 日経BP社

著者紹介

宮田 明(みやた あきら)

研究開発部 第八研究部 研究一課に所属。

music演人の研究開発を経て、現在クラウド基盤技術の開発に従事。

大久保 英幸(おおくぼ ひでゆき)

研究開発部 第三研究部 研究二課に所属。

music演人の開発を経て、現在生体信号の解析に関する研究開発に従事。

清水 晃(しみず あきら)

研究開発部 第八研究部 研究一課に所属。

music演人の開発を経て、現在サイクルスポーツ向けWebサービスの開発に従事。

城崎 康夫(しろさき やすお)

研究開発部 第八研究部 研究一課に所属。

music演人の開発を経て、現在サイクルスポーツ向けWebサービスの開発に従事。

新宅 貴行(しんたく たかゆき)

研究開発部 第八研究部 研究一課に所属。

music演人の開発を経て、現在サイクルスポーツ向けWebサービスの開発に従事。

福島 章雄(ふくしま あきお)

研究開発部 第八研究部 研究三課に所属。

music演人の開発を経て、現在スポーツ向けセンサーの開発に従事。