

# 携帯電話ソフトウェア開発の取組み

## Approaches to Development of Cell Phone Software

### あらまし

携帯電話の高機能化や高性能化に伴い、携帯電話のソフトウェア開発は、難易度が高くなり、品質確保の難しさや、開発コストの増大、進化スピードへの対応が求められている。

本稿では、それらの課題に対して富士通が携帯電話のソフトウェア開発において、実際に利用している手法を中心に取組みを紹介する。さらに、今後確実に到来する携帯電話のソフトウェアプラットフォームを中心とした開発環境の変化の中で、富士通を含めた携帯電話メーカーは、どのようにオープンプラットフォームを活用し、どのような取組みを行っていくべきなのかを論じる。

### Abstract

As cell phone handsets have come to have more enhanced functions and performances, so the level of difficulty of developing software for them has increased. This poses challenges including the difficulty of ensuring quality, increased development costs and addressing the speed of evolution. In this paper, we present approaches to the development of cell phone software mainly the techniques that Fujitsu actually uses to address these challenges. Furthermore, we discuss how Fujitsu and other cell phone handset manufacturers should make the most of open platforms and put to great use cell phone software platforms in the course of changes in the development environment, which are certain to take place in the future.



長内伊理一（おさない いりいち）  
ソフトウェア開発統括部 所属  
現在、NTTドコモ様向け携帯電話の  
ソフトウェア開発に従事。



吉澤尚子（よしざわ なおこ）  
アプリプラットフォーム開発統括部  
所属  
現在、携帯電話のアプリケーション  
ソフトウェアプラットフォームの開  
発に従事。

## まえがき

携帯電話のソフトウェア開発はソフトウェア母体規模の巨大化、機種を追うごとに増大する開発規模、複雑化により、開発難易度が非常に高くなってきている。

富士通は、こういった状況の中で様々な取組みを行い、QCD（Quality/Cost/Delivery）を考慮して開発を進めてきた。

しかし、LTE（Long Term Evolution）、オープン化、グローバル化といった世の中の流れに応じていくには、現状の開発のやり方では限界にきており、さらなる高機能化、低コスト化を推進していかなければならない。また、近年、携帯電話ソフトウェアプラットフォームを取り巻く環境が大きく変わり、オープン化が進んでいる。携帯電話メーカーは、このオープンプラットフォームをどう活用していくかを考えることが急務となっている。

本稿では、現状のソフトウェア開発における取組みの一例と今後の高機能化、低コスト化を図るための新たなプラットフォームの活用について紹介する。

## ソフトウェア開発の現状

昨今の携帯電話は、非常に高機能なものになっており、それに伴ってソフトウェアも巨大化、複雑化が進んでいる。

NTTドコモ様（以下、NTTドコモ）のmova端末

時代のソフトウェア規模と比較すると既に3倍以上の規模になっている。OS、ミドルウェア、通信制御の一部を除いて、ほぼ自社で開発しているため、毎機種の開発量も数十万行の開発となっており、増加する一方である（図-1）。搭載機能に関して各メーカーで特徴を出すために世の中のトレンドに合わせて、いち早く新たな訴求機能を実装することで熾烈な開発競争を繰り広げている（表-1）。

このような状況の中、ソフトウェアの開発は難易度が高くなってきており、QCDに応えるために次章で述べる取組みを行ってきた。

## QCDへの取組み

通常、ソフトウェアの開発とハードウェアの開発は並行で進められる（図-2）。

各開発工程の中ではQCDに応えるための取組みを行っており、その一例を紹介する。

表-1 機能搭載の変遷

年	項目
1998～2001	i-mode, カラー液晶, Java, カメラ
2002～2004	指紋認証, GPS, SDカード, 手書き認識, 歩数計
2005	FeliCa, iTunes, PDFリーダー
2006	ヨコモーション, WMA
2007	ワンセグ
2008	WMV, タッチパネルUI, 顔検出, 運動量計
2009	手書き入力, ゴルフ診断

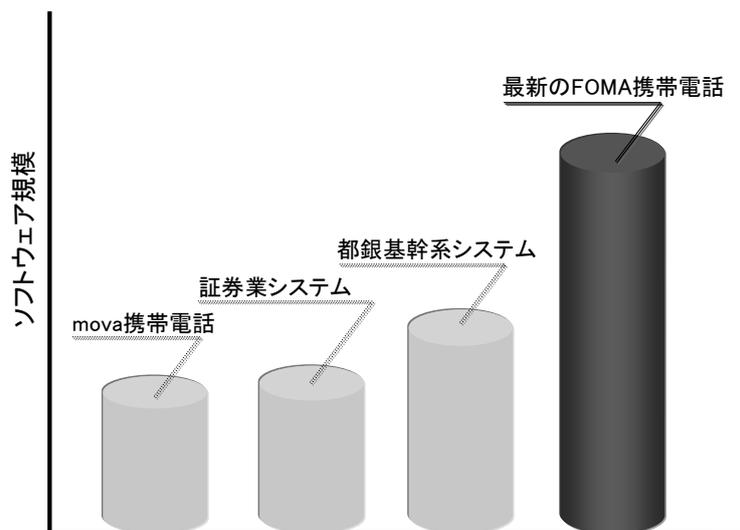


図-1 ソフトウェア規模  
Fig.1-Scale of software program.

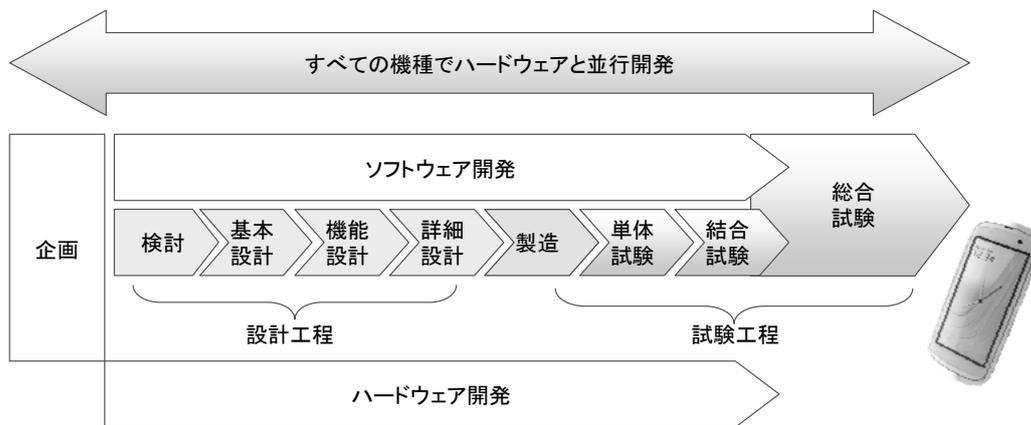


図-2 開発工程  
Fig.2-Development process.

(1) ターゲットレス試験環境

ソフトウェアの開発で試験工程の前半はハードウェアも安定していないため、ターゲットレス（携帯端末実機）での試験が必要になる。富士通は、ハードウェア、ドライバ以外をパソコン上で擬似できる環境を構築し単体試験，結合試験，および総合試験で発生した問題解析の効率化を図っている（図-3）。

(2) U管理図を用いた科学的品質分析

ソフトウェアの品質の分析には、主にバグ密度（プログラム1キロステップあたりのバグ件数）を利用している。しかし、開発規模に応じてバグ密度だけでは正常に分析できない場合（開発規模が小さいと、1件のバグのためにバグ密度の変動幅が大きくぶれ、指標から大きく外れたが問題があつて外れたのかどうか分かりにくい）があり、開発規模に応じた精度の高い品質分析のためにU管理図を導入した。管理図における中心線（バグ密度平均）、管理限界（正常の上限/下限）は、以下に示す数式で決定される。

$$\text{平均} : \bar{u} = \frac{\sum c_i}{\sum n_i}$$

$$\text{管理限界} : \bar{u} \pm 3 \sqrt{\frac{\bar{u}}{n_i}}$$

c:バグ数, n:規模

本管理限界は一般的には3シグマ限界と呼ばれ、限界から外れる確率は約0.3%である。「稀にしか発生しないことが起きることは異常である」という考



図-3 ターゲットレス試験  
Fig.3-Target-less test.

え方から、管理限界を外れたデータが示しているプロセスは何らかの異常があるはずと考える（図-4）。

(3) 試験工程における移動平均法の活用によるバグ収束時期の科学的予測

株価傾向予測などで使われる一般的な統計的手法である移動平均法を活用し、過去14機種バグ収束までにかかった日数を基に相関を調査した。

その結果、移動平均法の短期（5日）/長期（20日）バグ平均におけるデッドクロス発生時の「バグ件数」と以降の「バグ収束までの期間」に相関があり、精度の高い評価完了日が予測できるようになり、評価の過不足を防止し、試験の効率化が図れた（図-5）。

(4) 直交表による試験効果

多機能化、処理の複雑化に伴い、複合試験，競合

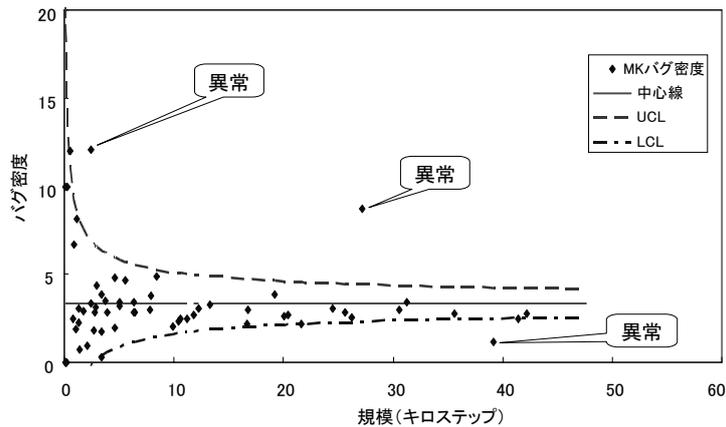


図-4 U管理図管理例  
Fig.4-Control case by U-control chart.

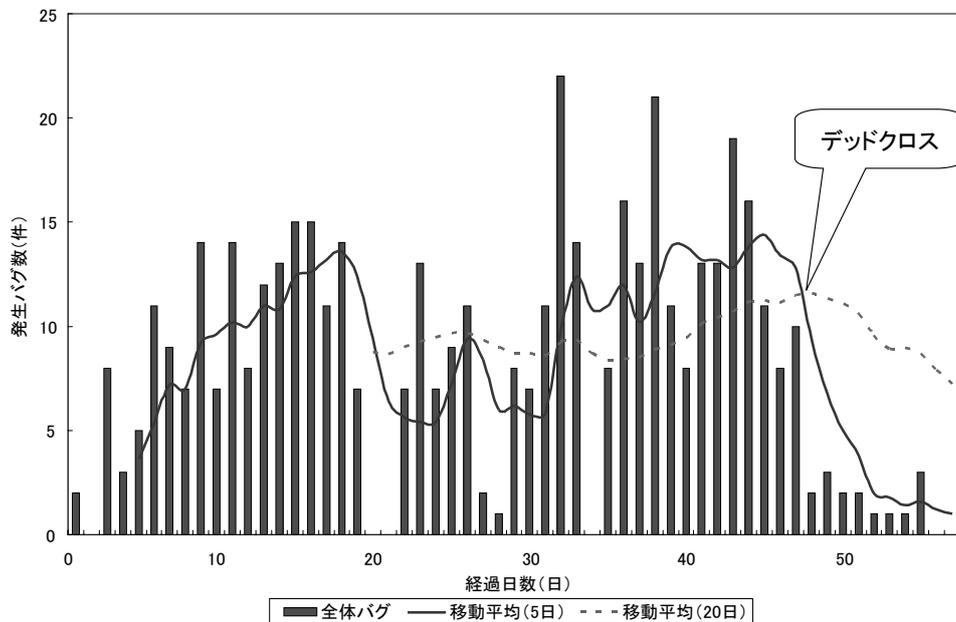


図-5 移動平均法の例  
Fig.5-Case of moving-average method.

試験もすべてのパターンを実施すると天文学的な項目数になり、現実的ではない。当初は開発者の経験から、組合せの抽出を行っていたが、スキルによってはムラや偏りが発生し、バグの流出が多かった。こういった偏りをなくし、効率良くバグを検出する目的で直交表を採用した(図-6)。直交表とは、実験計画法や品質工学(タグチメソッド)で使用される組合せ表で、パラメータの影響度を明らかにし、効率的に最適なパラメータ条件を見つけることを目的とした手法である。

(5) アプリケーションプラットフォームの採用

ソフトウェア開発コストの増加に対し、NTTドコモ固有機能と日本市場向け機能の両者を搭載した付加価値の高いミドルウェアを含むプラットフォームをMOAP<sup>®</sup>(Mobilephone Oriented Application Platform)として商品化しソフトウェア開発コストの低減を図った(図-7)。

(6) 課題

今後、LTE、オープン化、グローバル化などの世の中の流れに対応していくに当たって、相当な開発量が予想される。

現状の手法での開発でQCDを保っていくには限界にきており、高機能、低コストで開発可能な新たなプラットフォームの活用を考えていかなければならない。

新たなプラットフォームの活用に関しては後述する。

激変する携帯電話ソフトウェアプラットフォーム環境

過去2年間で、携帯電話を取り巻く環境で最も大きく変わったものの一つに、携帯電話のOSを含むソフトウェアプラットフォームの環境が挙げられる。2007年11月にGoogleが発表したAndroidがその一例である。以下、携帯電話業界におけるソフトウェアプラットフォームの動向をまとめる。

それまでの携帯電話ソフトウェアプラットフォームの常識を覆して、Googleが「Open & Free (オープンソースと無償化)」で携帯電話開発市場へ乗り込んだ。これを迎え撃つ形で、その約半年後にNokiaが携帯電話OSベンダのSymbian社の買収とともに、Symbian Foundation (SF) の設立を発表した。その間もLiMoやLiPSなどのモバイルLinux推進団体の再編も次々と起こり、携帯電話のソフトウェアプラットフォームは、iPhoneやWindows Mobileにも刺激を受け、「オープンプラットフォーム」という形で、現在に至るまで、進化し続けている(図-8)。

さらに、それまで日本市場においては、OSの上にキャリア共通のミドルウェアを構築し、その上の

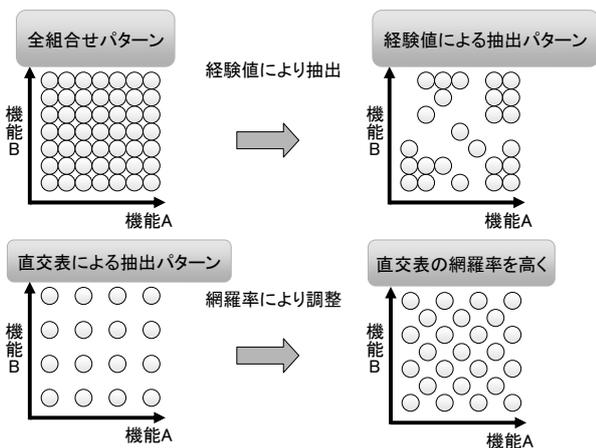


図-6 直交表の組合せ例  
Fig.6-Combination case by orthogonal table.



図-7 MOAP (Mobilephone Oriented Application Platform)  
Fig.7-Mobilephone Oriented Application Platform (MOAP).

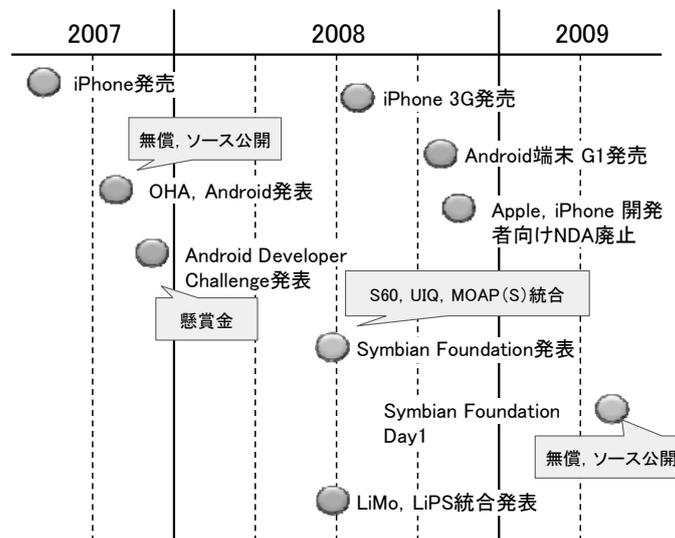


図-8 ソフトウェアプラットフォーム動向  
Fig.8-Trend of software platform.

アプリケーションは各携帯電話メーカーで実装するといった、言わば「共通ミドル型」のソフトウェアプラットフォームが主流であった。これに対し、オープンプラットフォームは、OSからミドルウェア、アプリケーションを含む「完結型」のプラットフォームを提供するが、この「Open & Free+完結型」のプラットフォームへの変遷は、日本の携帯電話のソフトウェア開発の仕方に大きなインパクトを与えつつある (図-9)。

## 携帯電話ソフトウェア開発の考え方の変化

これまで、一般的に日本における端末開発では、第2世代から第3世代の初期までは、キャリアが要件仕様を提示し、キャリア要件と独自機能のすべてを携帯電話メーカーが作り込んで端末を仕上げる方法で、ものづくりを行ってきた。

その後、第3世代もその普及に伴い端末機能とキャリアサービスなどの複雑なキャリア仕様が増えて、開発規模が膨らんできた。そのころから、キャリア要件のソフトウェアはキャリアが提供し、携帯電話メーカーはそれを組み込み、独自機能を追加して

端末を作り上げる形態に変わった。

それが今、競争の激化のために常に進化し続けなければいけない状況下で、莫大なソフトウェア開発費用の抑制の必要性とともに、キャリアに閉じた世界から、インターネットやクラウドといった外の世界とのシナジーを目的としたグローバル化が急速に求められている (図-10)。

すなわち、オープンプラットフォームが持つグローバル性をいかに使いこなすかがポイントとなる。さらに、オープンプラットフォームは、コミュニティによって進化・発展する形態が多く、オープンプラットフォームを使いこなすということは、取りも直さず、開発費をかけずに公開されているものを使うという意味で開発費の低減につながる。すなわち、オープンプラットフォームの積極的な利用は、グローバル化と開発費低減の一石二鳥の効果を生む。

しかし、ここで国内の携帯電話メーカーは壁に突き当たる。「ガラパゴス」と揶揄されてきた日本の携帯電話へのオープンプラットフォームの適用は簡単ではないからである。国内で一般的に普及している携帯電話 (いわゆる、キャリアフォン) が、海外の

	OHA	Symbian Foundation	Microsoft	LiMo	Apple
アプリケーション	Android				iPhone
ミドルウェア		Symbian Foundation	Windows Mobile	LiMo	
OS	Linux			Linux	MacOS X

図-9 ソフトウェアプラットフォーム構成とけん引主体  
Fig.9-Structure of software platform and leading organization.

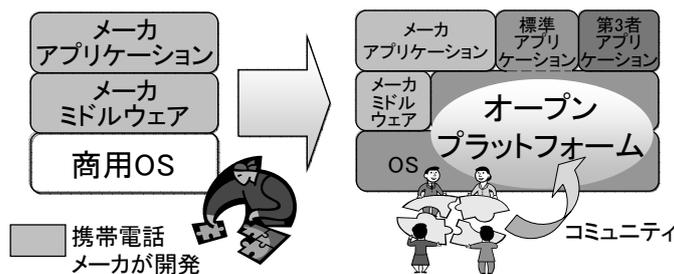


図-10 ソフトウェア開発の変化  
Fig.10-Change of software development.

ものと違いすぎるからである。例えば、ユーザインタフェース一つをとっても、“on hook”はアプリケーションの終了、“clear”は前に戻る、といった日本のキャリアの努力により統一されてきた動作が、海外のそれとは違うのである。また、メニュー構成も日本の携帯電話のものは階層が深く複雑すぎて、海外では受け入れられない。携帯電話メーカーにとっては、キャリアフォン向けには、オープンプラットフォームの必要な部分（グローバル性や付加価値機能など）をいかにうまく抽出し、これまで積み上げてきた膨大なキャリア要件のソフトウェア資産と統合していくかが、かぎなのである。

一方、スマートフォンは比較的ハードルが低い。すなわち、まだ日本においては普及が進んでいないため、ユーザエクスペリエンスが確立されておらず、市場がオープンプラットフォームの仕様に対して寛大で受け入れやすいからである。

国内市場においては、当面は、キャリアフォン、スマートフォンと2本立てでものづくりが進むと予想するが、iPhoneやAndroid端末に見るように、国内外の携帯電話メーカーがスマートフォンにも力を入れているため、国内市場でもスマートフォンが台頭してくると思われる。

携帯電話メーカーは、キャリアフォンやスマートフォンにオープンプラットフォームの恩恵をそれぞれにあった方法でうまく利用しながら、開発を進め

る必要がある。

### オープンプラットフォームの活用と活性化

オープンプラットフォームの活用は、グローバル性や共同作業によるコストシェアに対してメリットがある。しかし、反面、その活用は商品開発における差別化を困難にする。それは、オープンプラットフォームは「誰でも使える」から差別化が難しいというだけでなく、差別化しようとする標準が崩れ、いわゆるフラグメンテーションを起こすことがあるからである。

オープンプラットフォームと一口に言っても、そのオープン度合いによって様々な形態がある。アプリケーション開発者向けの開発環境（SDK）のみが提供されているものもあれば、SDKに加え、プラットフォームのソースが公開されているものもある（図-11）。

もちろん、オープン度合いが高ければ高いほど、フラグメンテーションの可能性が高くなるので、オープン度合いが高いプラットフォームでは、コミュニティはそれをルールやプロセスを設けて回避する。一方、iPhoneやBlackBerryなどの垂直統合モデルでは、自社でハードウェアおよびプラットフォームを作り、アプリケーション開発環境のみを提供することにより、フラグメンテーションは発生しない。オープンコミュニティにおけるフラグメン

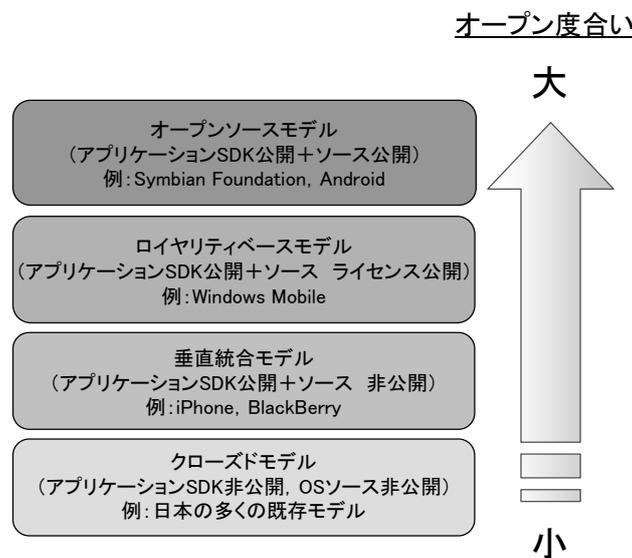


図-11 オープン度合い  
Fig.11-Level of openness.

ーションの回避手段は、それを標準インタフェースとして位置付けることである。

携帯電話メーカーやプラットフォームベンダは、差別化に必要であるが不足している機能セットやインタフェースセットをコミュニティへのコントリビューションという形でフラグメンテーションを回避する努力をするべきである。

オープンプラットフォームの魅力の一つはコミュニティによる進化スピードであるが、我々のような直接的な利用者は、その恩恵を継続的に受けるために、コミュニティの活性化に寄与していく必要がある。

携帯電話メーカーとしては、今後、以下の取組みが重要となる。

- (1) ソフトウェアベンダが魅力あるアプリケーションを市場（アプリケーションマーケット）に供給するために、オープンプラットフォームの機能（性能を含む）を確実に実装すること。
- (2) 標準に準拠していく方向が必要であり、必要に応じ、インターネットやハードウェアの標準インタフェースを提案、定義し、実装していくこと。
- (3) コントリビューションによるオープンプラットフォームの機能充実促進やフラグメンテー

ーションの回避に努力すること。すなわち、コミュニティエコシステムのプレーヤとなること。

### む す び

携帯電話のソフトウェアプラットフォームの激変の中で、携帯電話メーカーは変化への対応を求められている。

すなわち、これまでは、キャリア要件を中心に、そこにメーカー独自機能を追加する形で商品を作り上げてきたが、今後はキャリアに閉じた世界から外の世界とのシナジーを目的としたグローバル化が必要になってきている。

したがって、Symbian FoundationやAndroidといったオープンプラットフォームを、そのグローバル性などの必要部分を見極めてうまく利用し、いかにオープンに適した差別化の実装ができるかが開発のかぎとなる。また、オープンプラットフォーム自体の機能アップや性能改善を加速し、継続的に恩恵を受けられるようにするため、携帯電話メーカーは、同時にエコシステムの一人のプレーヤとして、コミュニティの活性化を行っていくべきである。

富士通もこれらの点を視野に入れ、携帯電話ソフトウェアプラットフォームの変化に対応していきたい。