2107CocoaReport
2021.7 ブログ：「接触確認アプリ COCOA からの教訓」を読んで、の詳細
（→ http://www.1968start.com/M/blog/index2.html#2107b ）

「接触確認アプリ COCOA からの教訓」を読んで

中所武司

■対象論文

情報処理学会の最新号(2021-07-15)［情報処理,62(8),384-392］：
特別解説「接触確認アプリ COCOA からの教訓 」

■本解説論文の読書のきっかけ

ＣＯＣＯＡの不具合については、すでに下記の２件のブログで言及した：

2021.4　接触確認アプリ「COCOA」の不具合の報告書を読んで
http://www.1968start.com/M/blog/index2.html#2104b

2021.2　新型コロナ接触確認アプリの不具合を４カ月放置
http://www.1968start.com/M/blog/index2.html#2102

上記４月のブログでは、厚労省の報告書へのコメントとして、
『結局、プロジェクト管理が全くされていなかったという一言に尽きる』と述べた。

その前の上記２月のブログでは、テストの不十分性、不具合報告放置の問題点に加え、プログラム構造（モジュール分割）が保守容易な構成でないのではないかと指摘した。

本解説では、現場の状況が具体的に述べられ、上記４月のブログ内容を裏付けるように、**ずさんなプロジェクト管理の実体が明らかにされている。**

■本解説の抜粋とコメント（→★の部分）

【まえがき】

・本解説の筆者は 2020 年 4 月から接触確認アプリの導入に関し，有志での議論に参加し，
有識者会議のメンバとして，また途中から政府 CIO 補佐官として，
接触確認アプリの導入を支援してきた．
・本稿では COCOA の開発と運用について，どのような課題があったか振り返る．

【接触確認アプリ導入の経緯】

・（2020 年）3 月頃：日本での接触確認アプリのリリースが話題となる。
・4 月：コロナテックチームが，内閣官房コロナ室と民間との協力関係の枠組みとなる．
・5 月：厚生労働省が調達する方向に方針転換が行われた．

【なぜ Exposure Notification API を採用したのか】

・Apple と Google が発表した接触通知技術（Exposure Notification API）について、
特に iPhone のシェアが高い日本において iOS で確実に動作することと、
十分なプライバシー水準を確保する必要があることから，この利用を決定．

【プライバシーを優先して断念した動作の追跡】

・5 月：政府の方針を受けてコロナテックチームの下に有識者会議を組織化した。
プライバシー保護のため、一意の識別子を用いたアプリケーションの動作監視はしないと決めたため、アプリの不具合の早期検出の手段はなくなった．
・5 月 26 日：アプリ仕様書とプライバシーの評価を公表
・5 月 27 日：厚生労働省は HER-SYS を委託していた業者に、接触確認アプリの開発および 7/31 までの運用保守を委託（HER-SYS の開発・運用保守に係る契約の追加契約），その他の 4 社への再委託を承認．
・9 月：切り分けが難しい複数の不具合への対策として、プログラム中でログを出力し，利用者がログの内容を確認した上でメールにて送信する仕組みとした。

【再委託先事業者の選定経緯】

・5 月：厚生労働省調達の方針決定の時点で、予算要求・調達手続きを行う必要があり、リリースは 7 月以降となることが想定された．
・一方で、政治的にも緊急事態宣言後の 6 月にはリリースすることが望まれていた．
・そこで、HER-SYS の執行残を使って COCOA を構築するのが，リリース最短の方法だった．
・開発着手までの時間的猶予がなく，既存契約の変更などで対応せざるを得なかった．

→★早期開発を最優先して、機能面、予算確保、業者選定で妥協したとのこと
（★ソフトウェア工学以前の情けない話）

【初期リリース段階でのテストは十分だったか】

・5 月 25 日：総理が，個人情報はまったく取得しない，安心して使えるアプリを，6 月中旬を目処に導入する予定と発表した．
・6 月：契約後 2 週間で，テスト仕様書・テスト報告書の提出は時間的に無理。
リリースを延期すべきことは明らかだった。

・本番環境でテストするほかないならば，対象を限定してリリースすることが望ましい．
　プレビューとして限定的にリリースする提案も行われたが，その場合、
　総理が 6 月中旬にリリースすると発言していたことと齟齬が生じてしまう．
・**6 月 19 日**：利用者の陽性登録までには時間差があるので、リリースした．
　一方、品質を担保できないので， 1 カ月近くはプレビュー版として提供し，
　広く利用者からのフィードバックを受けることとした．
・リリース直後から数多くの不具合が見つかった．
　＊HER-SYS との連携が動作せず，陽性登録を行うことができなかった．
　＊接触判定の重み付けがダミーの値で，過剰に接触通知を出す設定だった．
　見つかったバグの性質としては，結合テストの途中段階の品質だった．

→★テスト不十分のままリリースしたら、予想通り数多くの不具合が見つかった。
　（★テストを利用者に頼むという、ソフトウェア工学以前の情けない話）

【9 月の Android 版リリースにおける不具合の発生要因】

・**８月～９月**：接触リスク判定に関係なく，軽微な検知についても接触通知が多発し、
　検知した接触の最大リスク値が閾値よりも大きい場合だけ通知するように改修．
　この際、iOS と Android の Exposure Notification API の実装に差異があり、
　Android では接触通知を行わなくなった。
→★共通仕様（ＡＰＩ）の実装の差異が、外部仕様の差異になったということは、
　共通仕様の記述に不備があったということだ。

【実機テストの実施を阻んだ開発環境の制限】

・iOS と Android の API 実装の差異は、実機テストを行っていれば，検出できたはずだが、
　開発版と接続できる HER-SYS の接続検証環境がなく，実機テストできず．
・この場合，以下のいずれかを構築してテストすべきだった。
　＊HER-SYS のように振る舞う**スタブ**
　＊実機検証用のバックエンド
　＊簡単にテスト用の診断キーを登録できる，テスト専用のアプリ
・10 月に実機テストの環境が構築済みだったが，なぜか 12 月のリリースで
　実機テストを十分に行わなかった．

→★スタブ利用については、古くからのテスト技術として常識だった。
（参考：**30 年以上前の拙著**）「プログラミングツール」、昭晃堂 (1989) p.152
　5 章「テスト」の 5.3「テスト実行」の 5.3.2「ツール」の(1)「**環境模擬機能**」の中で、
　図 5.17「モジュールテストのための**環境模擬機能**」を掲載し、
　『未作成の下位モジュールを模擬するスタブ定義機能』と記述している。
　　http://www.1968start.com/M/book/progToolCH5.pdf

また、
　拙著：「ソフトウェア工学」、朝倉書店（2014） p.71
　7章「テスト」の7.1.2「テスト工程」の中で、
　『被テストモジュールから呼び出すモジュールを模擬するスタブ』と記述している。

【GitHub 上の issue が見落とされた背景】

・COCOA のソースコードは9月から GitHub 上で公開されている
・契約上，委託業者が GitHub 上の issue を確認する義務はなかった．
・指摘の多くは技術的なもので，厚生労働省の職員が内容を理解することは難しかった．
・事前に責任者を決めていなかったので，適切な対応ができなかった．
・運用保守を引き継いだ業者に瑕疵担保責任はなかった．

→★要するにプロジェクトマネージャーのいないプロジェクトだった。
→★不具合の放置については、下記のブログ参照：
　2021.2　新型コロナ接触確認アプリの不具合を4カ月放置
　　http://www.1968start.com/M/blog/index2.html#2102

【厚生労働省の調査報告書と，今後の再発防止に向けて】

・**2021年4月16日**：厚生労働省は不具合の発生経緯の調査と再発防止の報告書公表．
・再発防止策として：
　＊開発当初から外部システムとの結合テストを実施するための環境を整備しておき、改修のたびに一連の動作検証を行うこと，
　＊連携先システムや API 等の改修による影響を前提とした契約とすること，
　＊政策判断を担う管理職が IT リテラシーを持って，IT 専門家を活用していくこと．
・迅速にサービスを構築して提供するには，発注者が品質に対する責任を負うこと．
・現状では、受入テストも形骸化し，品質管理はベンダ任せとなっている．
・COCOA の失敗を端緒として，硬直的な調達制度や要員管理，管理態勢にメスを入れて，環境変化に柔軟に適応できる開発体制と，品質管理に必要なガバナンスを確立すべき．

→★今回の失敗の教訓と同様のことは、すでに e-Japan 以来の20年間に数多くあった。
　失敗の教訓を生かせなかったのは、根深い構造的な問題によると思われる。
→★上記報告書については、下記のブログ参照：
　2021.4「接触確認アプリ「COCOA」の不具合の報告書を読んで」
　　http://www.1968start.com/M/blog/index2.html#2104b
　『結局、プロジェクト管理が全くされていなかったという一言に尽きる』
と述べている

以上