



<スクエアfreeセミナー>
30分でわかるユーザビリティ設計の基本
～ITシステムにおいて重要性を増す非機能要件への取
り組み方～

ブライテクノ株式会社 中西 良明
2023年9月28日



発表者略歴

中西 良明 NAKANISHI Yoshiaki
ブライテクノ株式会社 代表取締役

- 経歴

- 大阪大学 情報工学分野修士課程卒業
- 松下電器産業(株)に入社し、インターネット・情報セキュリティ・携帯電話関連技術の研究開発に従事
- マカフィー(株)では携帯電話・スマートフォン向けのセキュリティソフトの研究開発を担当
- ソニー・エリクソン・モバイルコミュニケーションズ(株)でAndroidデバイス開発プロジェクトに参画
- 2010年ブライテクノ(株)を創業

- 趣味

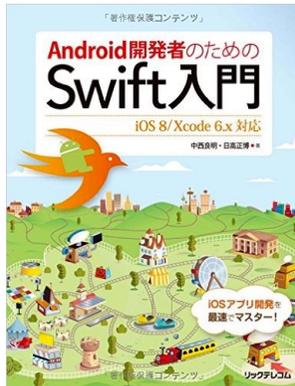
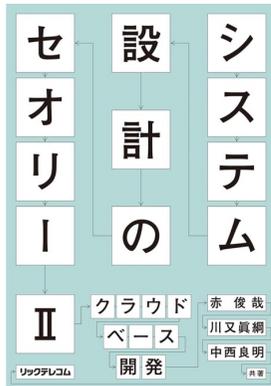
- 謎解きゲーム



nakanisi@brighttechno.com
X(旧Twitter) @chun_ryo



著書・講演・教育実績



システム設計のセオリーII (リックテレコム) **New!**

クラウドベース開発のシステム設計における理論と手順

<https://www.amazon.co.jp/dp/4865943633>

Android開発者のためのSwift入門 (リックテレコム)

SwiftによるiOSアプリ開発の入門書

<https://www.amazon.co.jp/dp/4897979838>

Effective Android (インプレスジャパン)

Androidエキスパートによる高レベルなアプリ開発技術の指南書

<https://www.amazon.co.jp/dp/4844335340>

Product Manager Bootcamp (Google Japan)	セキュリティとAndroidアプリ品質ガイドラインのセッション
ワイヤレス技術セミナー (リックテレコム)	Android・iOS・ウェアラブルに関するセッション
Droid Kaigi 2016 (一般社団法人Droid Kaigi)	Android 6.0の省電力機能に関するセッション
Tech Institute (早稲田大学・角川アスキー総研)	Androidのセキュリティに関する授業を担当

会社概要



ブライテクノ株式会社 / Brighttechno, Inc.

代表取締役	中西 良明
創立年月	2010年10月
所在地	東京都千代田区岩本町2-13-6 ミツボシ第3ビル 703 ・都営新宿線「岩本町」駅 徒歩3分 ・JR線「秋葉原」駅 徒歩5分 ・東京メトロ日比谷線「小伝馬町」駅 徒歩6分
電話番号	03-5809-1814
URL	https://brighttechno.com/
業務内容	ソフトウェア開発およびコンサルティング ・ Android/iOS アプリケーション開発 ・ IoTソフトウェア開発 ・ Webアプリケーション開発 情報セキュリティコンサルティング



今日お話しすること

- ▶ 機能要件・非機能要件について
- ▶ 標準準拠とUX向上の関係について
- ▶ 実践するための壁と対策について



機能要件と非機能要件（ざっくり）

▶ 機能要件

- システムによって実現したい目的を満たすために必要なもの
 - 使いにくくても目的を達成できるだけの機能があれば充足できる

▶ 非機能要件

- システムの使いやすさ、性能、保守のしやすさなど
 - 業務システムなら、利用者の生産性向上を達成するための能力
 - コンシューマ向けシステムなら、利用者がファンになってくれるための能力



非機能要件は軽視してよいのか？

- ▶ 性能要件が示されていなかったら？
 - タスク実行(例: 1件の帳票の登録や更新)時にユーザーを5秒待たせてもOK
 - 1000人の従業員による利用なら、1件あたり5000秒(約1時間20分)の時間が無駄に費やされる

- ▶ 保守要件が示されていなかったら？
 - 2週に1回、12時間のメンテナンスが発生するシステムでもOK
 - 平日に止めることはできず、休日にエンジニアの稼働が発生する
 - コンシューマ向けサービスが月に1回メンテナンスに入ってもOK
 - 機会損失による収益への影響、メンテナンスによって利用できなかったユーザーの離脱の発生などが考えられる



非機能要件は軽視してよいのか？

- ▶ 使いやすさについての指標がないとどうなるか？
 - スマートフォンで電話番号入力のフォームをタップした時、日本語入力のキーパッドが表示される
 - ユーザーが数字キーパッドに切り替える手間をさせられる
 - PCで画像やファイルアップロードする時に、ファイル名に使えない文字があるとして拒絶される
 - ユーザー側からすると理不尽なエラーをつきつけられる



非機能要件は軽視してよいのか？

- ▶ 結論：軽視してはいけない
 - 社外に開発を依頼する場合、社内で開発する場合のいずれであっても、要求仕様には十分に検討した機能要件と非機能要件を記載すべき
 - 「○○機能が欲しい」だけでなく、使い勝手、性能、保守性なども示す
 - とはいえ、システム・PC・スマートフォンの専門家ではないビジネス部門が、適切な機能要件と非機能要件を示すことは難しい



- 社内外の専門家に頼る
- リファレンス（お手本）、標準ライブラリ、OSS、クラウド(SaaS等)、ローコード・ノーコード開発ツール、共通開発環境を活用する
 - 作る開発・作らない開発を検討する上で重要



リファレンス（お手本）の活用

- ▶ お手本になるもの（モバイルUIの場合）
 - 公式アプリ（例：標準メールアプリなど）
 - 公式アプリは基本的にデザインガイドラインに準拠している
 - ユーザーは公式アプリのUIに慣れている
 - 利用者の多いアプリ
 - LINE、メルカリ、Xなど
 - デザインガイドラインに準拠しつつ、独自の使いやすさを追求している
 - デザインガイドライン
 - AndroidはMaterial Designのガイドライン
 - <https://m3.material.io/>
 - iOS/iPadOSはHuman Interface Guidelines
 - <https://developer.apple.com/design/human-interface-guidelines>

技術者向けの英語資料のため
ハードルは高い



リファレンス（お手本）の活用

▶ 発注者

- 標準アプリ、有名アプリ、デザインガイドラインの存在を意識する
 - 開発者からの提案が適切かどうかを判断する基準を持つ
 - RFPに非機能要件にそれらの情報を提示できるようになる

▶ 開発者

- 標準アプリ、有名アプリ、デザインガイドラインについて知っておくべき
 - 少なくとも必要になった時に最新情報を参照・検討し顧客に提案できる
- 実装方法について知っておくべき
 - 標準ライブラリでできるのか、有名なOSSはあるのか、など



標準ライブラリ・OSSの活用

- ▶ 多くのアプリ・システムで実装されている機能のほとんどは、標準ライブラリやOSSとして提供されているもの
 - スクラッチから自作するよりも、非機能要件も考慮された洗練されたものを利用できる
 - スマートフォンに表示されている画面を、下方方向にスワイプして離すことでロードする機能
 - 大手テック企業が自社内で開発したUI部品をOSSとして提供する場合あり
 - OSバージョンアップなどを考慮して開発されている
- ただし、開発者がメンテナンスをやめるリスクはあるので、その場合の方針をあらかじめ決めておく
 - 自社でメンテナンスを継続する
 - 乗り換え先を探す、など



クラウドの活用

- ▶ いろいろなシステムが必要とする機能はXaaS、ノーコード・ローコードツールとして提供されている
 - ユーザー管理のシステムおよびUI
 - IDaaS
 - ノーコード・ローコードシステム
 - MAツール、CRM
 - スマートフォンアプリ構築
 - 業務システム 構築, etc.
- ▶ 独自開発との比較
 - 独自開発・運用コストとの比較（開発期間への影響も考慮する）
 - サービス提供を中止されるリスク、価格上昇リスクの評価



クラウドの活用（ユーザー管理）

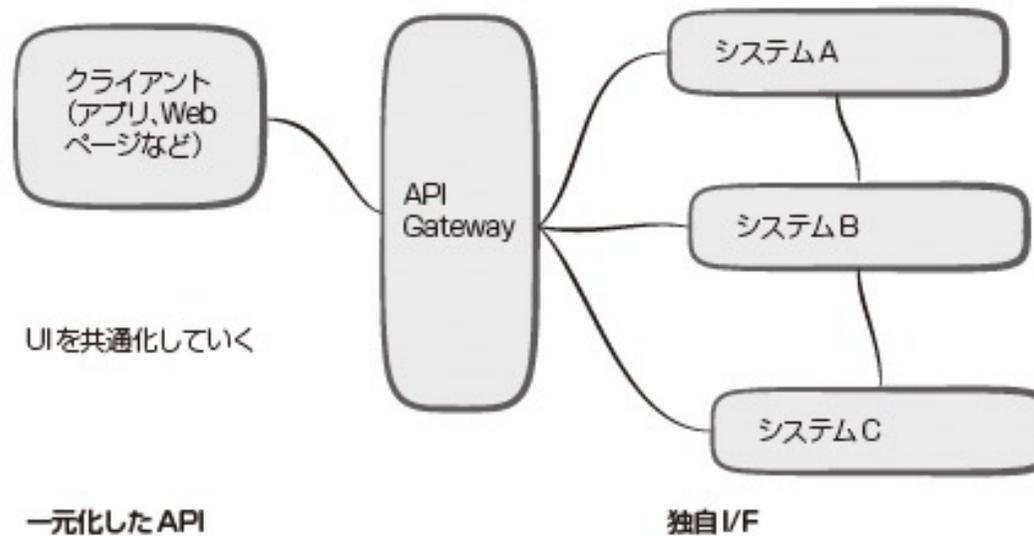
- ▶ ユーザー管理機能をオンプレミスやIaaSで持つか、IDaaSを利用するか
 - コンシューマ向けなら、安価なソーシャルログイン機能の利用が第一候補
 - ユーザー登録・ログインのUIまで含めてサービスから提供され、セキュリティの責任を軽減できる
 - たとえば、Firebase Authenticationの場合MAU 49,999まで無料
 - 業務用システムのID管理の場合、コストを加味して検討する
 - アカウントごとに月額数百円かかるが、高度なユーザー管理機能が提供される
 - たとえば、1000アカウントで1アカウント500円/月かかる場合
 - 月額500,000円の利用料が発生
 - オンプレミスやIaaSで実現した場合、ソフトウェア導入費(開発費等)、セキュリティ対策、保守・運用費等を元に計算する



構成要素の交換に備える

- ▶ クラウドのサポート期間、提供期間はコントロールできない
 - API Gatewayなどの仕組みを用いることで、最小限の影響で構成要素を交換できるように備える

図6-5 改善後のシステム概念図



システム開発のセオリーIIより抜粋



ここまでのことを実践できるか？

▶ 発注者（ユーザー企業）

- ビジネス部門の方が、どんなクラウドシステム、OSS、標準機能があるかなどを細かく知って対応するのは現実的には不可能
- 社内IT部門でも、日々の業務を抱えている中で最新技術情報をキャッチアップすることは難しい
- 技術コンサルティングができるR&D部門などを持っているのが理想的

▶ 開発者（受託開発会社）

- 顧客からの要求で非機能要件について示されていない場合に、検討して提案することが望ましい
- 最新技術動向などをキャッチアップできていることが期待される



「社内IT部門に頼る」は難しい？

- ▶ 受託開発会社が接する案件情報の大部分が、ビジネス部門だけで考えられたものに見える
 - 社内に専門部隊がまったくない（中小企業ではよくある）
 - IT部門が保守に特化した人材で構成されている
 - ふんわりした要件から稟議のために概算金額を知りたいだけ
 - ビジネス部門とIT部門の風通しが悪い



- まず自社内でどこまでできるかを正確に把握した方がよい



作る開発・作らない開発検討の前に

- ▶ DX(デジタルトランスフォーメーション)でどう変化すべきかをまず考える
 - 社内IT部門が保守に特化していてDXの検討に適していないなら、外部専門家に頼る
 - できれば自社サービスなどに誘導しない、独立性が高い専門家を利用できるとよいがなかなか難しい
 - 保険アドバイスサービス事業者が保険代理店業になっているように、IT化促進の補助金・助成金を活用を提案する事業者なども代理店ビジネスであることが多い



最新技術動向に強い専門家はどこにいる？

- ▶ 残念ながら、受託開発会社には非常に少ない
 - 最新情報を追いかける優秀な専門家は大手テック企業に引き抜かれがち
 - 大手テック企業は、プラットフォームの最新情報に精通した部隊を抱えていることが強さになることを理解している
 - 高い技術力や素養を持った技術者は、業務で最新技術に触れられる環境を強く求めている
- ▶ 社内で育つ環境を作るのが理想、出来ないならコンサルティングを利用する
 - 大手テック企業所属エンジニアと、副業マッチングプラットフォームで繋がるのも一つの手



生成AIに要件定義を作ってもらうことは可能か？

- ▶ 生成AIに指示して要件定義をまとめることはできそう
 - NTTデータが生成AI使うシステム開発手法を全技術者に展開、人月型契約が見直しに(日経クロステック)
 - <https://xtech.nikkei.com/atcl/nxt/column/18/02423/073100046/>
 - ChatGPTに要件定義をお願いしたらハンパなかった(DevelopersIO)
 - <https://dev.classmethod.jp/articles/gpt-requirement-definition/>
- ▶ 画期的・独創的なシステムについては適さないか？



生成AIでシステム開発できるか？

- ▶ かなり使える技術ではあるが、うまく使いこなすためにも基礎的な知識を要する
 - 定型的なアプリであれば、要求を伝えて生成しやすい
 - 定型的なアプリは学習元データに含まれていることが多い
 - ノーコード・ローコード開発を置き換えていくかもしれない
 - 生成AIの機能で開発をしていくこともある程度はできるが、できあがったものが本当に正しいか、セキュリティ上の問題が無いかなどのチェックは必要
- ▶ 少なくとも現時点では、生成AIを活用する良い技術者を利用することが最適解

参考情報



- ▶ 次ページ以降で紹介



共通開発環境

▶ Flutter

- Googleが主導するAndroid/iOS/Web/PC/Mac向けの共通アプリ開発環境
- Javaに似たDart言語を用いて開発できるため、AndroidやiOSアプリ開発者にとって使いやすい

▶ ReactNative

- 元々Web開発向けプラットフォーム
- JavaScriptで開発ができるため、Webアプリ開発経験者にとって使いやすい

▶ .NET MAUI

- Microsoftが主導する共通開発環境
- C#とXAMLで開発できるため、Windowsアプリ開発経験者にとって使いやすい



モバイル向けノーコード・ローコード開発

- ▶ テンプレート選択 + カスタマイズ開発サービス
 - 事業者が提供する機能の組み合わせだけで開発するのであれば、高速かつ比較的安価に高機能アプリをリリースできる
 - 独自のカスタマイズが多いと、スクラッチからの開発と大きな差はでにくい（逆転する場合もあり得る）
- ▶ FlutterFlow
 - Android/iOSなど向けのアプリをWeb上の設定のみで作成できる
 - 安い、ただし日本語説明なし
 - 生成されたアプリのFlutterソースコードをダウンロードできる
 - ダウンロードしたコードを元にカスタマイズ開発もできますが、かなり難解なコード（コメントなど無し）が出力されるので、現実的ではなさそう



ご静聴ありがとうございました