

利用状況把握から始める UX 課題分析手法の提案

- Web システムの利用品質の向上を目指して -

Proposal of a UX analyzing method starting with understanding contexts of use

- To improve a quality in use for web systems-

主査 : 金山 豊浩 (株式会社ミツエーリンクス)
副主査 : 三井 英樹 (Weblysts.com) 村上 和治 (東京海上日動システムズ株式会社)
リーダー : 志賀 愛弓 (TIS 株式会社)
研究員 : 清水 有子 (日本電子株式会社) 水野 智仁 (株式会社ヴィッツ)

研究概要

ソフトウェアの利用品質を高めるためには利用状況の把握が必要である。しかし、開発現場では利用者の情報は推測や伝聞に留まり、機能要件の充足が重視される傾向にある。利用状況を把握する方法として UX 手法が提唱されているが、実践している開発現場は少ない。その原因として予算及び期間の制約がある。

そこで本分科会では、開発者のみで低コストかつ短期間で UX を適用する、開発現場のためのソフトウェア利用品質向上手法を提案する。現行ソフトウェアの「利用状況の把握」と「UX 五階層による分析」を軸とする。本手法を実際のプロジェクトに適用したところ、ソフトウェア課題の抽出と分析において有効性を確認したため併せて報告する。

Abstract:

In order to improve a software quality in use, it is necessary to know user usage and behavior. However, in many software development projects, software requirements are more important than usability. They are too busy to gather user information, so it remains a matter of speculation or hearsay. Though UX methods for understanding context of use are proposed, but only a few projects are using it. The reason is because there are budgetary and time constraints in software development projects. To solve this issue, we propose a process which improves a software quality in use for developers; a method to apply by a low cost and in short span without UX analysts. This method is based on "understanding contexts of use" and "the UX Five-stage analysis". We applied the technique to an actual project and confirmed the validity on identifying and analyzing software problems.

1. はじめに

1.1 研究の背景と UX 手法導入における問題点

ソフトウェア開発の現場では対象ソフトウェアの利用状況を把握せずに開発を行ったり、推測や伝聞による情報でプロジェクトを進めたりしがちである。その結果、ソフトウェアが完成した後にその使いにくさが発覚するという失敗に陥る場合がある。ユーザーに望まれるソフトウェアを作るためにはユーザーを取り巻く状況やユーザー自身を知る必要がある。そのための方法の一つとして UX 手法が提案され、効果も確認されている[1][2]。しかし、実際の開発現場において UX 手法は浸透していない。その理由を開発に携わる研究員からヒアリングし分析したところ、

第4分科会 (FastUCD チーム)

- (1) 手法の適用には UX の専門スキルが必要であり、社内にスキルを持つ人材がいない点
- (2) 予算が限られており UX 専門家の参画や設備投資にコストが掛けられない点
- (3) 開発期間がタイトで新たなプロセスを導入する余裕がない点

といった課題があげられた。

1.2 研究の狙い

本稿では、予算や開発期間の制約内で利用品質を向上させることを目的とし、UX 専門家なしに開発者のみで UX を適用する手法を提案する。提案手法では、既存ソフトウェアの利用状況を簡易リモートユーザーテスト(以下 FastUT と呼ぶ)にて把握し、課題を洗い出す。FastUT はクラウドソーシング [解説: 付録 A] サービスを活用する。次に、洗い出した課題を UX の観点で五階層に分類し、対応の優先順位をつけて検討 (UX 五階層分析と呼ぶ) する。これら一連の流れを「F2 法 (Fast User Test and Five-UX Analysis method)」と呼ぶ。この手法を開発者のみで実施し効果が得られるか、かつ省力化できるかを検証する。

2. 従来型手法と提案手法について

利用状況の把握にはエスノグラフィー、ユーザーインタビュー、現場観察、ユーザーテストなどが用いられる [3]。なかでもよく使われるのはユーザーテストである。そこでまず従来型ユーザーテストの概要を示す。次に F2 法によって対応できる範囲を説明する。

2.1 従来型ユーザーテスト

従来型のユーザーテスト (以下、従来型 UT と略す) では UX の専門家が、テスト計画 (ゴール・課題のヒアリング、テスト設計)、参加者のリクルーティング、テスト実施 (インタビュアーによる進行、撮影・録音)、結果分析と報告 (レポート作成など) まで実施する [4]。UX 専門家に求められるスキルはユーザーテスト設計力、インタビュー力、分析力など多岐に渡り [5]、開発者が同じスキルを身に付けて代行することは難しい。また、撮影・録音機材の他にアイトラッキングシステムなどの専門機材が必要な場合もある。

2.2 従来型 UT と F2 法

従来型 UT の作業と F2 法の作業を対応付けたのが次の表である。「サービス側で対応」の箇所はクラウドソーシングサービスの中でカバーされており開発者の作業は不要、または簡略化している。「開発者が実施」の箇所は開発者が行う必要のある作業を示す。このように本手法では、従来型 UT で行っている部分を簡易に実施できるよう、サービスを利用し簡素化している。次章より手法の詳細について説明する。

表 1 従来型手法と F2 法の対応

従来型 UT の工程		F2 法		手法詳細の章立て
		サービス側で対応	開発者が実施	
計画	目的の設定	-	○	3.1 (1)対象ソフトウェアの戦略・要件設定
	確認事項の設定	○	-	-
	測定項目の設定	○	-	-
	テスト参加者のプロフィール設定	○	○	3.1 (2)参加者選定
準備	参加者選定用質問票作成	○	○	3.1 (2)参加者選定
	テスト参加者募集	○	-	-
	タスクシナリオの作成	○	-	-
	テスト用質問票作成	○	○	3.1 (3)テスト質問票設計
	テストのワークスルー	-	-	-
	パイロットテスト	-	-	-
実行	進行	-	-	-
	録音・撮影	○	-	3.1 (4)テスト実施
結果分析と報告	問題の特定	※	○	3.1 (5)「発見点」のリストアップ
	解決案の提案	※	○	3.2 UX五階層分析
	報告書作成	-	-	-

【凡例】 ○ : 対応有、 - : 対応なし、 ※ : オプションサービスが用意されている

3 提案手法 (F2 法)

3.1 簡易リモートユーザーテスト (FastUT)

ユーザーテストをリモートで行うクラウドソーシングサービスを利用し、UX 専門家なしに、かつ自前で設備を用意せずユーザーテストを実施する。今回の研究では(株)ポップインサイトの協力のもと、ベータ版サービスである「ユーザテスト Express(beta 版)」[6]を利用し、サービスの狙いや利用方法をヒアリングした上で検証を行った。その結果、同サービスに対して後述の分析を組み合わせることで、開発者のみで UX を適用する開発に活かせることが分かった。次項より詳細な進め方を説明する。

(1) 対象ソフトウェアの戦略・要件設定

通常のソフトウェア開発では要求定義・要件定義工程においてソフトウェアを作る際の「狙い・目的」が定義されている。その中で根本目的にあたる内容を【戦略】、目的達成のための機能要件・タスクに近い内容を【要件】として定義する。【戦略】と【要件】は関連付いている点に注意する。「狙い・目的」が不明な場合は、ソフトウェアの【戦略】及び【要件】を運営者や企画者にヒアリングし定義する。

(2) 参加者選定 (「ユーザテスト Express(beta 版)」の場合)

ソフトウェアを使用してもらいたい参加者の選定条件を設定する。(1)で設定した【戦略】を元に、ターゲットユーザーがどのような人かを検討する。条件は様々に設定可能だが、簡易に行うため下記の設問を穴埋めする形とする。

表 2 参加者選定条件

選定条件	○○○	を	×××	したことがある人
------	-----	---	-----	----------

例 1: 「ビールのキャンペーン」を「スマホから応募」したことがある人

例 2: 「生命保険の解約」を「検討」したことがある人

その後、設定した条件に合う候補者名簿からテストしてほしい参加者を選択し依頼する。

(3) テスト質問票設計 (「ユーザテスト Express(beta 版)」の場合)

テスト設計は専門的な知識が必要な作業である。これをテンプレート[付録 C]に従って設計することで、開発者のみで作成可能とする。また、対象ソフトウェア(サービス)の種類によって追加のテンプレートを利用する。

(4) テスト実施 (「ユーザテスト Express(beta 版)」の場合)

候補者名簿から参加者を選択し、遠隔地にいる参加者にテストの実施を依頼する。参加者はインターネット経由で対象ソフトウェアへ接続する。従来型 UT では UX 専門家が進行役を務めるが、FastUT では質問票に沿って参加者が操作するため、進行役は不要である。試験の様子(画面と発話)は参加者がテスト動画として保存する。

(5) 「発見点」のリストアップ

テスト動画を閲覧し、発見点(参加者の言動や挙動)を一覧化する。問題点だけではなく良い点も全て記録することが重要である。次工程の分析で良い点も把握し、ソフトウェアの現状を理解するためである。

3.2 UX 五階層分析

前述の方法にて発見点を洗い出せるが、そのうち問題点は全て並列で優先順位付けしていない。そのため、対象ソフトウェアを総合的な観点からどのように改善すべきか判断がつかない。また、予算や期間の制約により全ての問題点に対応できない場合がある。そこで発見点の整理と対応の優先順位付けを目的として、下記の通り分析する。

(1) 発見点の分類

3.1 の FastUT にて挙げた発見点を Jesse James Garrett が提唱するユーザーエクスペリエンスの要素[付録 B]に基づいて五階層に分類する(表 3, 付録 F 参照)。前述(3.1)の(1)の【戦略】【要件】を除いた、残りの【表層】【骨格】【構造】の三階層へ分類する。

表3 UX五階層の分類

No.	階層	内容	優先順位	発見点の例
1	表層	色, サイズ, 画像など見た目に関わる部分	低 ↑	・リンクと説明文との区別がつかない ・アイコンの意味がわからない
2	骨格	ページ内の構成に関わる部分 (どこに何を配置するかなど)		・関連情報が離れて配置されており読みにくい ・情報やボタンがファーストビューになく見落とす
3	構造	画面遷移, システム全体のメニュー構成や機能に関わる部分		・ある情報へアクセスするための階層が深くたどり着けない ・必要な機能や情報がない
4	要件	ソフトウェアで実現したい事	↓ 高	(発見点には上がらないためなし)
5	戦略	ソフトウェアの根本目的		

(2) 発見点の関連付け

次に, 3.1 でヒアリングした【戦略】と【要件】を起点とし, 発見点を表3の下から上に進む形でツリー型に紐付けする. これをUX五階層樹形図と呼ぶ(図1).

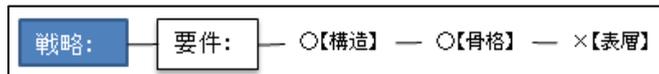


図1 UX五階層樹形図

発見点は, 良い点と問題点を視覚的に区別しやすいよう, 先頭にO×やアイコンなどで表現する. ここでは良い点をO, 問題点を×で表現した. どの要件にも紐付かない発見点は「要件:その他」に紐付ける(図2の①). また, 階層間の分類が歯抜けの箇所は「(なし)」の箱で紐付けする. 例えば【要件】→【構造】→【骨格】のうち, 【構造】にあたる発見点がなく【骨格】の発見点が挙がっている場合, 【構造】を(なし)で紐付ける(図2の②).

加えて, 類似の内容や, 同じ機能及び画面に対する発見点を紐付けし, 各発見点の関連を図示する. 但し異なる要件で同じ機能や画面にコメントが表れた場合は離れた箇所であって良い. 同じ内容について良い点・問題点に意見が分かれた発見点は近接させる(図2の③).

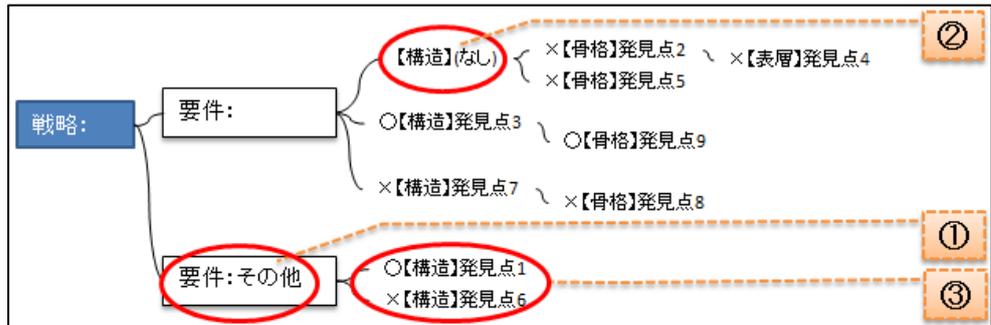


図2 UX五階層樹形図の例1(発見点の関連付け)

は離れた箇所であって良い. 同じ内容について良い点・問題点に意見が分かれた発見点は近接させる(図2の③).

(3) 分析

作成した樹形図を元に分析結果をグループ化し, システムの良い点・問題点に大別して(図3の①列), 問題点には改善案を追記する(図3の②列). また, 改善案が前述「UX五階層分類」のどの層にあたるかを合わせて検討する.

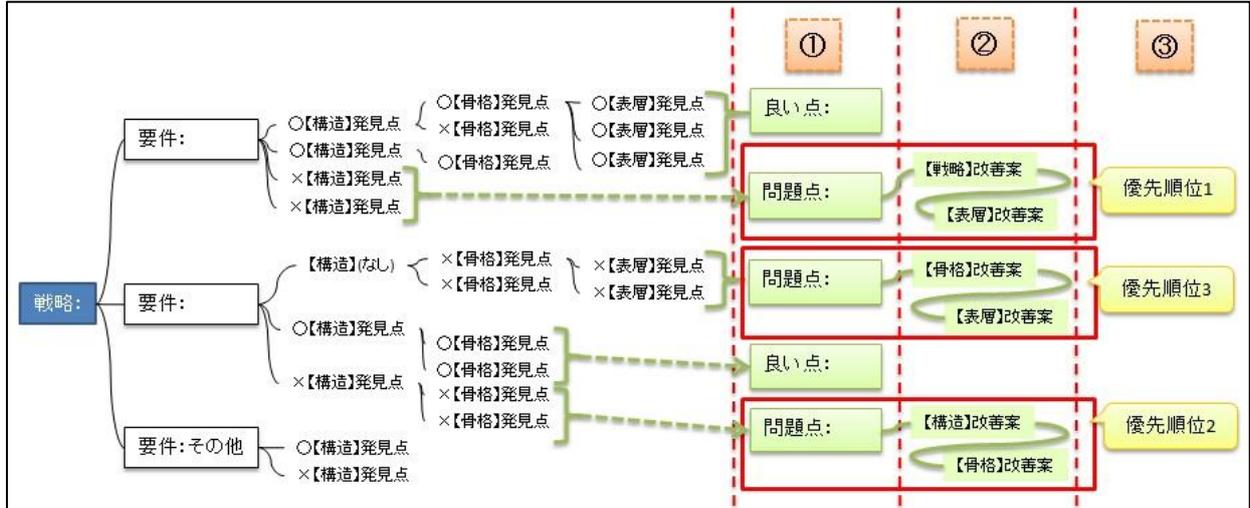
(4) 優先順位の付与

大別した問題点(図3の①列)の改善案(図3の②列)が【戦略】に近いほどその問題点の優先順位を高く設定する(図3の③列). したがって, 改善案の紐付きによっては【表層】の改善案も一概に優先順位が低いとは言えない. 例えば【表層】の改善案が【戦略】

第4分科会 (FastUCD チーム)

の改善案と紐付く場合は、その【表層】の改善案は【戦略】に影響する改善案と捉えて優先順位を上げて対応すべきだからである。

図3 UX五階層樹形図の例2(分析, 優先順位付与)



なお、複数の問題点に対応する改善案の階層が並列になった場合の優先順位は、下記の式で算出した数値(P)の値の大きい方から順位を高く設定する。

$$P = N \cdot \sum_{i=1}^5 F(i) \cdot k(i)$$

N: 出現人数
 i: UX五階層の分類(表3のNo.と同一)
 F(i): 階層iの改善案の重み
 k(i): 階層iの改善案の数

■ UX 五階層分類の重み (F(i))

複数の問題点の改善案が UX 五階層分類のうち同じ分類だった場合に、改善案の階層の割合を加味した重み付けを行う。例えば、問題点 A = {改善案:【構造】4件,【骨格】1件}, 問題点 B = {改善案:【構造】1件,【骨格】4件} の場合、問題点 A の方を高い優先順位に設定するためである。重み関数 F(i) は対象ソフトウェアの特性に合わせて関数を変えるのが望ましい。今回は F(i)=i とする。

■ 出現人数 (N)

出現人数とは、問題点グループに含まれる問題点を指摘した参加者の総数を表す。1名の参加者にしか挙げられなかった問題点と、複数の参加者で類似の問題点が挙げられた場合がある。前者はその参加者固有の問題の可能性があり、後者は他の参加者でも同様の問題が挙がる可能性が高い。この点を勘案し出現人数を加味して計算する。

3.3 F2法のメリット

従来型 UT と F2 法について比較し F2 法が優位な点を次の表に示す。

表4 従来型 UT と F2 法の比較

比較観点	従来型 UT	F2 法
専門家の参画	必要	不要
設備購入 (記録機材, Web カメラ等)	必要	不要
テスト質問票作成	難しい	易しい
参加者リクルーティング	必要	不要
実施期間	1~2 か月	2 日~1 週間程度
費用	数百万円	数万~十数万円程度

上記のように、従来型 UT と比べて F2 法は短期間・低コストで実施可能である。また、専門家の参画は一切必要とせず、テスト質問票もテンプレートに従えば、全て開発者のみ

第4分科会 (FastUCD チーム)

で行うことができる。

3.4 F2法のデメリット

F2法ではUXの専門知識がない人でも実施可能なようにテストをテンプレート化している。また、進行役が同行しない、参加者がリモートからアクセスするなどの違いがある。そのため、従来型UTと比べると以下のデメリットがある。実施の際には留意されたい。

表5 F2法のデメリット

比較観点	従来型 UT	F2 法
参加者の行動に応じた質問の掘り下げ	可能	不可
参加者の表情	見える	見えない
参加者の IT スキル	制限なし	一定程度必要※1
対象ソフトウェアのオンライン接続	不要	必要

※1 参加者はオンラインでソフトウェアにアクセスしテストするため、ITスキルが全くない人物が実施することは困難である

4. 適用結果

4.1 音楽団体のウェブサイトへの適用

F2法を実際のプロジェクトに適用し、その効果について報告する。対象は音楽団体のウェブサイトにおけるチケット販売である。

対象とするウェブサイトは16ページで構成されており、大別すると右記の通りである。

表6 適用対象サイトの構成

種別	ページ
チケット販売	5
ブログ	2
その他の静的ページ	9
合計	16

4.2 FastUTの実施

次の表に示す通り、各参加者一人あたり約10~15件、計53件の発見点が挙げられた。内、約半数が問題点である。FastUTによって発見されたこれらの問題点の多くは、サイト運営者が認識していない内容であった。

表7 発見点数 (参加者毎)

参加者	発見点数(件)		
	良い点	問題点	小計
A	6	10	16
B	11	4	15
C	6	6	12
D	4	6	10
合計	27※1	26	53

※1 表8は異なる参加者による同一観点をまとめた数字のため、表7の合計と一致しない

表8 発見点数 (UX五階層分類毎)

UX五階層分類 ※2	発見点数(件)	
	良い点	問題点
表層	4	1
骨格	7	11
構造	12	14
合計	23	26

※2 要件・戦略は発見点とはならない

4.3 UX五階層分析の適用

発見点をUX五階層分析にて分類し、課題が集中する箇所を整理したところ5個の問題点にまとめられた[図4(部分), 付録D(全体図)]。結果の概要を次の表9に示す。分析により、対象サイトの全体構成や雰囲気、情報量は良い評価である点、構造上の課題が多い点が明確になった。また、サイト戦略見直しの必要性がわかった。

図4 UX五階層分析結果 (部分)



表9 対象サイトにおける分析結果 (概要)

分類	優先順位	UX五階層分類					分析結果
		戦略	要件	構造	骨格	表面	
良い点	-	-	-	○	-	○	全体の雰囲気・印象は総じて良い
	-	-	-	○	○	-	チケット・コンサートページの情報量は十分
問題点	1	○	-	-	-	-	サイトのコンセプト明確化が必要
	2	-	-	○	○	-	購入検討ユーザーが演奏会詳細にアクセスしやすい導線が必要
	3	-	-	○	○	○	座席表の選びやすさ、わかりやすさに課題
	4	-	-	○	○	-	購入途中の不安解消必要
	5	-	-	-	○	-	複数演奏会比較の観点漏れ

なお、五階層分類で【構造】に当たる問題点が3つ検出されている(優先順位2~4, 薄い網掛け)。これらの優先順位は、前述(3.2)の式で算出し、数値の大きい方から順位を高く設定する[付録E]。ここでは五階層分類の重みの値は付録Eの通り、5~1の値で設定した。階層の差をより重要視する場合は、重みの値を大きく取り対応する。

4.4 考察

4.4.1 発見点の抽出と分析

検証の結果、F2法によって多くの問題点を抽出できた。注目すべきは、開発者はユーザーの行動を推測してサイトを構築していたが、FastUTによって推測の誤りが明らかになった点である。一例を挙げると、開発者の想定では、ユーザーはどのチケットを購入するか既に決めている(=すぐにチケット購入ページへ進みたい)という思い込みがあった。しかし実際のユーザーは、どのチケットを購入するか検討する(=演奏会詳細を閲覧する)フェーズを重視していた

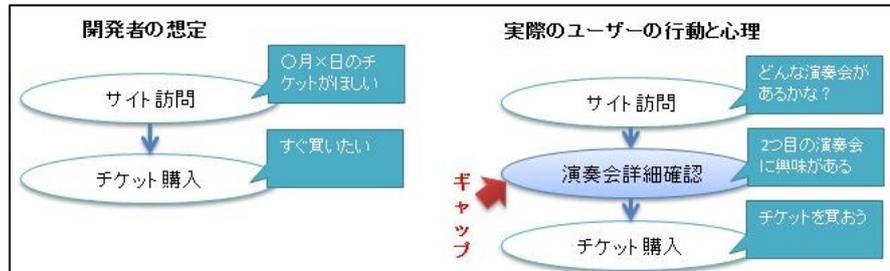


図5 開発者の想定とユーザーの行動のギャップ

開発者の想定では、ユーザーはどのチケットを購入するか既に決めている(=すぐにチケット購入ページへ進みたい)という思い込みがあった。しかし実際のユーザーは、どのチケットを購入するか検討する(=演奏会詳細を閲覧する)フェーズを重視していた

第4分科会 (FastUCD チーム)

[付録 G 参照]. 開発者はトップページからチケット購入ページへの導線を重視していたが、演奏会詳細への導線がわかりにくいという問題には気づかなかったのである。

また、FastUT は問題点が全て並列で挙がり、ソフトウェアが現状どういった状態なのか評価しにくい。しかし UX 五階層分析により、26 点の問題点が 5 つの問題点に大別でき、優先順位をつけて検討可能となった。

4.4.2 利用品質向上効果の検証

F2 法利用の有無による改善効果の比較を行った。F2 法による改善案(A)と、ユーザー調査を行わずに検討した改善案(B)の二種類を作成し、両者にユーザーテストを実施した。その結果、テスト参加者 4 名中 3 名が(A)の方が使いやすいと評価した。

4.4.3 適用工数と期間

表 10 提案手法の適用工数と期間 (適用例における実績)

適用例にかかった F2 法の作業内容と所要時間を表 10 に示す。F2 法の作業は 1 人日未満の工数で実施可能であった。前述の表 4 からわかるとおり、従来型 UT に比べ 10 分の 1 の期間で実施可能である。そのため、開発期間がタイトな場合も F2 法を導入可能と考える。

作業内容	所要時間	期間
FastUT 申込～ユーザー選定 (4 名分)	40 分	1 日間
動画閲覧, 発見点洗い出し (4 名分)	2 時間	1 日間
UX 五階層分析	3 時間	
合計	5 時間 40 分	2 日間

5. 今後の展望

本提案手法により、UX の知識なしにユーザーの利用状況を把握し、低コスト・短期間で既存ソフトウェアの問題点を挙げられることがわかった。今後は適用例を増やし五階層分類の重み付け数値の精度を向上させ、更なる効果の立証と手法の改善に努めたい。

また F2 法の課題として、FastUT は対象ソフトウェアにインターネットアクセス可能でないと実施できない点があげられる (限定的であれ、ネットワークに接続できれば実施は可能である)。そのため、閉じたネットワーク環境でしか利用できないソフトウェア、基幹業務系ソフトウェア等では利用できない点が課題である。今後は環境の制限があっても開発者が簡易にユーザーテストを行える方法を模索していきたい。

6. 参考文献

- [1] 第 21 年度ソフトウェア品質管理研究会第 4 分科会, 開発現場における UCD アプローチ実践の課題, 日本科学技術連盟, 2005
- [2] 第 28 年度ソフトウェア品質管理研究会第 4 分科会, 本当に望まれるソフトウェアを開発するための UXD (User eXperience Design) 活用検証, 日本科学技術連盟, 2012
- [3] 第 29 年度ソフトウェア品質管理研究会第 4 分科会, システム開発における利用者視点欠乏症の簡単自己診断と処方箋一覧, 別紙 3, 日本科学技術連盟, 2013
- [4] SQuBOK 策定部会 編集, ソフトウェア品質知識体系ガイド (第 2 版), 2014
- [5] 人間中心設計推進機構, 人間中心設計専門家 コンピタンスマップ (2013 年改訂版), http://www.hcdnet.org/certified/docs/competence_map.pdf, 2013
- [6] (株)ポップインサイト, ユーザテスト Express, <https://usertesting.jp/express/>
- [7] 第 24 年度ソフトウェア品質管理研究会第 4 分科会, プロトタイピング手法の効果的な選択方法の提案, 日本科学技術連盟, 2008
- [8] WILLIAM SAFIRE, <http://www.nytimes.com/2009/02/08/magazine/08wwln-safire-t.html>, 2009
- [9] Jesse James Garrett (浅野紀予訳), The Elements of User Experience, http://www.jjg.net/elements/translations/elements_jp.pdf, 2000

付録A クラウドソーシングとは[8]

クラウドソーシングとは crowd + outsourcing の造語である。自社の従業員または他社にて実施していた業務を、インターネットを介して不特定多数の人々へ公募して外部発注する行為を言う。

付録B ユーザーエクスペリエンスの要素

「The Elements of User Experience : (日本語訳)ウェブ戦略としての「ユーザーエクスペリエンス」 —5つの階層で考えるユーザー中心デザイン—

J. J. Garrett 氏によって創られた概念である。【ユーザーエクスペリエンス】を高めるために、Web を設計する際に考慮すべき項目を構造化されています。Web は以下に示す二重性を持っており、この関係を意識しながら、開発ステップで検討すべき「戦略、要件、構造、骨格、表層」の5階層の一貫した流れの中で【ユーザーエクスペリエンス】を作り上げることの重要性を記述しています。二重性とは「ソフトウェア・インターフェースとしてのWeb」と「ハイパーテキスト・システムとしてのWeb」であり、それぞれ「プロセスに関わる各ステップにおいてタスクの遂行について人々がどう考えるか」と、「サイトがどんな情報を提供するのか、そして、その情報がユーザーにとってどんな意味を持つのか」という課題を持っています。 [7]

本概念はWebの説明として述べられているが、画面を持つソフトウェアについてはその構成要素がWebシステムと同様であり、本概念を適用可能である。

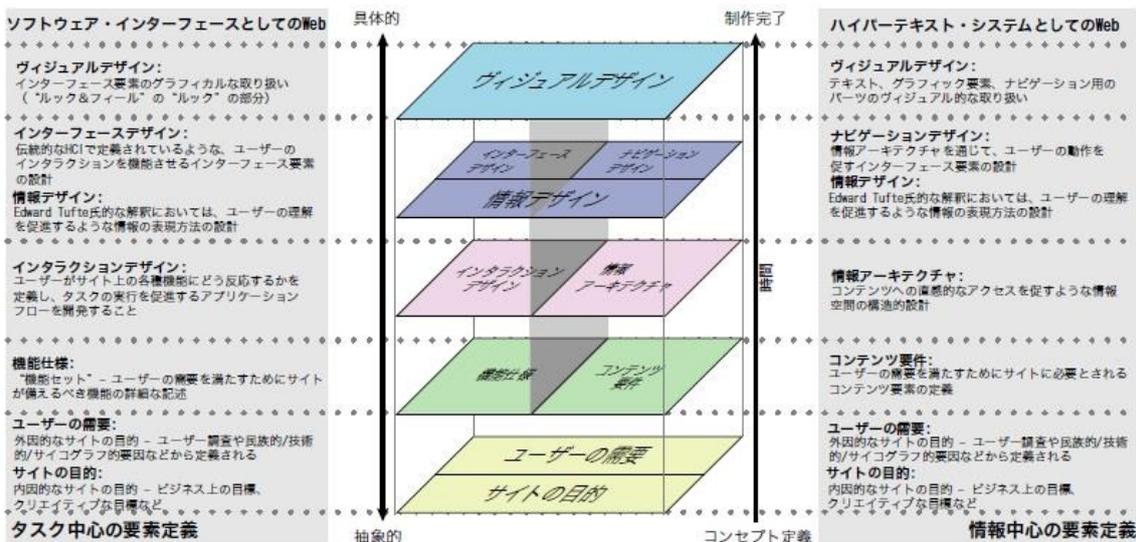
ユーザーエクスペリエンスの要素

Jesse James Garrett
jjg@jjg.net

30 March 2000

Translated by Noriyo Asano
日本語訳：浅野 紀子

Webが包含する基本的な二重性: Webは元来、ハイパーテキストをベースとした情報空間と考えられてきたが、ますます洗練されたフロント/バックエンド双方のテクノロジーの開発が進むにつれ、リモートなソフトウェア・インターフェースとしてのWebの利用も増えてきた。ユーザーエクスペリエンスを実現しようとする人々が、当初の適用範囲から逸脱するケースにまで各自のターミノロジーを適用しようとするにしがたが、上記のようなWebが包含する二重の性質と相まって多くの混乱を招いている。この文書が目指すゴールは、これらの用語のいくつかを適切な文脈に沿って定義し、これら多様な要素(エレメント)の背景にある関係性を明確にすることである。



この図は未完成です:
上図に描かれているモデルでは、ユーザーエクスペリエンスの開発途上での意思決定に影響するおそれがある二次的な(技術開発やコンテンツ開発の過程で表面化してくるような)検討事項は考慮していません。また、このモデルでは開発プロセスの記述も、ユーザーエクスペリエンス開発チーム内の役割の定義もしていません。あくまでも、現在の軌道におけるユーザーエクスペリエンスの開発に欠かせないキーとなる考慮事項を定義することがねらいです。

© 2000 Jesse James Garrett

<http://www.jjg.net/ta/>

第4分科会 (FastUCD チーム)

付録C 質問票テンプレート

※本テンプレートの内容は(株)ポップインサイトの許可の元、同社の「ユーザテスト Express」を参考に作成している

表 B-1 はどのようなソフトウェアであっても共通で使用する。ソフトウェアの特性に応じて、表 B-2 や表 B-3 を併用する。

表 C-1 質問票テンプレート (共通)

No.	質問	備考
Q1	あなたが「〇〇〇を×××したい」と思っていると仮定し、【対象ソフトウェア】を閲覧して検討してください。十分見たと感じたら、次の質問に進んでください。	「〇〇〇を×××したい」は 2.2.1 参加者選定時の内容と同一。対象ソフトウェアはトップページを指定する。
Q2	実際に「〇〇〇を×××したい」と仮定した時に、このサイト（ソフトウェア，サービス）の印象を教えてください。 {1：非常に悪い，2：悪い，3：良い，4：非常に良い}	
Q3	その理由を具体的に教えてください。	
Q4	サイト（ソフトウェア，サービス）についてよいと思った点を具体的に教えてください。	
Q5	サイト以外の点について良いと思った点を理由も含めて具体的に教えてください。	
Q6	サイトについて悪いと思った点，良くわからなかった点を具体的に教えてください。	
Q7	サイト以外の点で悪いと思った点，不安な点などを理由も含めて具体的に教えてください。	
Q8	過去に「〇〇〇を×××した」体験・経験を元に「こうしたらもっと」「もっとこうしてほしい」点を具体的に理由も含めて教えてください。	「〇〇〇を×××したい」は 2.2.1 参加者選定時の内容と同一。

表 C-2 質問票テンプレート (情報発信)

No.	質問	備考
Q9	先ほどのサイトで【特定のページを指定】は見ましたか？ {1：見た，2：見ていない}	サイト上重要な確認したいページを指定する。(製品詳細ページ，販売申込ページ等)。複数ページ設定してもよい。
Q10	Q9 のページ・サービスの印象を教えてください。見ていない場合は，見た上でお答えください。 {1：悪い，2：やや悪い，3：やや良い，4：良い}	
Q11	その理由を具体的に教えてください。	

第4分科会 (FastUCD チーム)

表 C-3 質問票テンプレート (EC サービス)

No.	質問	備考
Q9	購入に際し不安点や疑問点があれば調べてください.	
Q10	購入してみてください. 但し実際には購入せず, 確認ページまでで止めてください. また, 情報はコピーせずに手で入力してください.	個人情報はダミー情報を用意する.
Q11	購入前にどんな不安や疑問がありましたか?	
Q12	不安や疑問に対する情報の探しやすさはどうでしたか? {1 : 非常に悪い, 2 : 悪い, 3 : 良い, 4 : 非常に良い}	
Q13	その理由を具体的に教えてください.	
Q14	購入手続きのしやすさ (フォームの使いやすさなど) について, 印象を教えてください. {1 : 非常に悪い, 2 : 悪い, 3 : 良い, 4 : 非常に良い}	
Q15	その理由を具体的に教えてください.	

付録 E UX 五階層の同一分類における優先順位計算 (適用例)

分析結果	改善案の数 (k(i))					重み付け点数 F(i) × k(i)	出現人数 (N)	重み付け点数 × 出現人数 (P)	優先順位
	戦略	要件	構造	骨格	表層				
	UX 五階層分類の重み (F(i))								
	5	4	3	2	1				
購入検討ユーザーが演奏会詳細にアクセスしやすい導線が必要	0	0	3	2	0	13	2	26	2
座席表の選びやすさ, わかりやすさに課題	0	0	1	1	1	6	2	12	3
購入途中の不安解消必要	0	0	1	2	0	7	1	7	4

付録 F 五階層分類フローチャート

発見点を UX 五階層のうち三階層 (【構造】【骨格】【表層】) へ分類する際のフローチャートを下記に示す。分類に迷う場合の一助とされたい。



付録 G 改善前と改善後のユーザー操作の違い

【改善前：6ステップ】

①

② 購入ページにきてしまった (怒り顔) どの公演かわからない (怒り顔)

③ リンクがない (怒り顔)

④ ポスターが拡大しない (怒り顔)

⑤ ここかな? (疑問顔)

⑥ やっと見つけた! (笑顔)

【改善後：2~3ステップ】

①

② 詳細を確認できた (笑顔)

③ ポスターも確認できた (笑顔)