



# 最良のUXを実現するには？ UX初心者向けガイド

企業にとってユーザーエクスペリエンス (UX) は、もはや避けて通れない用語です。最良のUXは、「顧客の心にダイレクトに届く」という点で、ビジネスにおける他の側面では容易に構築できない貴重なチャンネルを生み出してくれるからです。わずか2文字の略語にしては大きすぎ、と感じるなら、アップルやテスラといった大きな成功を収めている大手企業が独自のUXを積極活用してきたことを思い出してみてください。圧倒的にポジティブなユーザーエクスペリエンスは、献身的な顧客を数多く生み、企業を熱狂的な人気を誇るブランドへと成長させてくれます。本ホワイトペーパーは、製品開発に関連したUXのすべてを詳しく解説します。UXとは何か、なぜ大切なのか、最良のUXを目指すために何ができるかといったさまざまな考察や基本的なベストプラクティスをご紹介します。貴社の製品で最良のUXを実現するのにお役立てください！



# 目次

UXとは? .....	3
目に見えないデザイン .....	4
ユーザー .....	5
UXと文化 .....	7
UXと感情 .....	7
初回ユーザーエクスペリエンス .....	8
UXとフロー .....	9
エクスペリエンス .....	9
デフォルトコンフィギュレーション .....	10
エラーへの対応 .....	11
インスタントフィードバック .....	14
UXとファネル .....	14
最も重要なのは「一貫性」 .....	15
UXとKISS .....	16
UXプロセスの構築する上で .....	16
デザインの反復 .....	17
経時的な進化 .....	18
UXテスト .....	18
まとめ .....	20

## UXとは？

UXについてよく知らない方もご安心ください。UXの正確な意味は今なお議論的であり、専門家の間でしきりに論じられていますが、ISOが先ごろついに、「製品やシステム、サービスの利用または予期された利用に起因する人の知覚や反応」という標準的な定義を策定しました。

### ISOによるUX定義の詳細

UXとは、製品やシステム、サービスの利用または予期された利用に起因する人の知覚や反応です。ユーザーエクスペリエンスには、ユーザーの利用の前中後に生じる感情や確信、嗜好、知覚、身体的／精神的反応、行動、成果のすべてが含まれます。ユーザーエクスペリエンスはまた、ブランドイメージやプレゼンテーション、機能性、システムの性能、相互作用的行動、インタラクティブシステムの支援的機能のほか、それに先立つエクスペリエンスがもたらすユーザーの内的／身体的状態や態度、スキル、パーソナリティおよび利用状況の結果として生まれるものでもあります。ユーザー個人の目的という視点から考えた場合、いわゆるユーザビリティには、このようなユーザーエクスペリエンスの知覚的および感情的側面が関わってきます。従ってユーザビリティの基準を用いれば、ユーザーエクスペリエンスのさまざまな側面を評価することができます。

UXデザイナーは、快適に利用できる製品を創造することによって、ユーザーのニーズに応えようとします。これは簡単なことではありません。なぜなら、エクスペリエンスは極めて多面的な概念だからです。個々人の知覚や振る舞い、動機、理解によって形成されるエクスペリエンスは、当然ながら一人ひとり違うものであり、同一のインタラクションから生じる印象もまた、一人ひとりで大きく異なるからです。さらにエクスペリエンスは、インタラクションに先立って形成されるユーザーの期待によっても大きく左右されます。たとえば、見た目が美しく品が良く、直感的に操作できるユーザーインターフェース(UI)を備

えた音楽配信サービスが、常にユーザーのニーズに応えられるとは限りません。目当ての曲が簡単に見つかれば、ユーザーはそのUXを高く評価しますが、好みの曲が見つからないことが何度も続けば、優れたUIでお粗末なUXを補うことはできません。



ユーザーインターフェース(UI)とは、人がソフトウェアアプリケーションやハードウェアデバイス(グラフィックUI、キーボード、マウス、ボタン、スイッチなど)を操作するための手段である。このようにユーザーと製品のインタラクションを支援するUIは、製品のUXに不可欠な一要素であると言える。

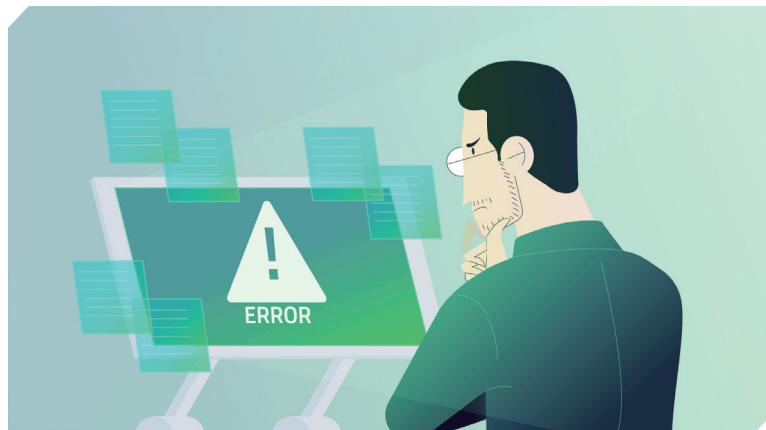
もう一つの難題は、UXはユーザーの感覚や感情に左右されるため、容易に測定できない特性があるという点です。最良のユーザーエクスペリエンスは、情熱や興奮、関心を呼び覚ますものでなければなりません。製品デザイナーが見た目の美しさやシンプルさばかりを追求すれば、UXに何の影響も与えられないどころか、マイナスの影響を及ぼすことさえあります。優れたユーザビリティだけで最良のUXが生まれるわけではないのも、そのためです。ユーザビリティ(製品の使いやすさ、使い方の身につけやすさ)はあらゆるUXの重要な一部分ですが、唯一の要素ではないのです。

むしろ、複雑なユーザーインターフェースを備えた製品がユーザーに難しい製品をマスターしたという達成感をもたらし、発見する楽しみを与え、高額な製品の価値をさらに高める要素、つまりUXのクオリティを押し上げる要素となる場合もあります。UXのクオリティはつまり、使う人によって変化す

るのです。さらには、ユーザーの知覚や動機、期待が常に一定ではない点にも注意が必要です。知覚や動機、期待は、置かれた環境に対するユーザーの認識や別の技術との出会い、競争や市場の力学、変化し続けるニーズといったさまざまな要因によって変化します。



UXのクオリティは、使う人によって変化する。



## 目に見えないデザイン

人は悪いデザインにはすぐに気づきますが、最良のデザインがユーザーの関心を引かない場合もあります。デザインの原則の多くは美しいものを作ること

に焦点を当てますが、UXのデザインにおいては、「目に見えないこと」が最も重要です。UXの始祖とも呼ばれる[ドン・ノーマン](#)は自著『[The Design of Everyday Things](#)』の中で次のように記しています。「良いデザインは私たちのニーズにぴったりとフィットし、デザインそのものを見えないものにして、私たちの注意を引くことなくその役割を果たす」。視覚的なデザインでありながら、「目に見えない」デザインもあります。デザインを簡素化することによって、過度に知覚に訴えるのを抑止し、ユーザーがタスクそのものに100%集中できるようサポートする手法です。

そのほかにも、見えないデザインによってニーズに自動的に応えたり、利用できるオプションをタイムリーに提示したりすることで、製品とユーザーのインタラクションを減らすこともできます。このように見えないデザインはさまざまな方法で、製品をユーザーの一部へと変えていきます。UXデザイナーはデザインに関する知識や製品知識、人間の特性、世界観というさまざまな要素を巧みに組み合わせて、見えないデザインを創造します。さらに、見えないデザインのこの「さりげなさ」は、優れた製品の背後にいる優秀なUXデザイナーの存在すら消し去るので

# ユーザー

ユーザーエクスペリエンスはユーザーを主体として開発する必要がある、という意見には誰もがうなずくでしょう。ただしここでもう一点、ユーザーが誰で、どのように製品を使うのかということも考えなければなりません。ターゲットオーディエンスのことをよく分かっているつもりでも、ユーザーの詳細なプロフィールを作成すればニーズをより上手に予測し、応えられるようになるはず。またユーザープロフィール(ペルソナとも呼ばれます)は、ユーザーの行動に関する疑問への回答を導き出す上でも有効です。

UXのユーザーペルソナは、さまざまな**テンプレート**を利用して作成できます。ただし最も正確かつ説得力のあるペルソナを作成するには、潜在的な顧客を綿密に調べ、アンケートやインタビュー、フォーカスグループなどを実施する必要があります。



**年齢**  
30歳

---

**性別**  
男性

---

**学歴**  
大卒

---

**職位**  
工場の保守監督者

## ユーザーペルソナ メンテナンス・マイク

**ニーズ**

稼働停止のリアルタイム通知機能。使いやすいレポートソフトウェア、統合的でシンプルかつ直感的なUI、自動化されたソリューション。

**現在の課題**

予知保全ができず、工場の稼働停止時間が長く、機器類が古いために、プロセスフローが複雑化している。

**目的と動機**

業務全体を監督し、保守担当スタッフを指導できる監督者になりたい。

**業務内容**

データを記録する。プロセスのトレンドを監視して記録を取り、業務の効率性を高める。規制関連のレポートを実施する。機器類の点検を行い、プロセスの欠陥や弱点を検知する。小規模な修理を行う。

**製品の使い方**

メーターや計器、指針盤を読み、詳細なプロセスチャートを作成する。機器類に必要な大規模修理に関するレポートを作成する。

**使用している／気に入っている他社製品**

Nintendo Switch、その他のゲームコンソール、iPhone、iPad。

**気に入らない他社製品(理由も)**

操作方法が分からないボタンや表示がたくさんある洗濯機。複雑で使い勝手が悪い。

ペルソナを作成するには、高度な調査と抜け目のない知見が不可欠です。ユーザーのニーズについて数百個もの質問をするのは生産的ではありません。ユーザーが自分のニーズを分かっていない、言葉で表現できない、ニーズに矛盾があるといった場合も考えられます。従って調査では、ユーザーの目的、その目的を現在どのように達成しているか、どのような課題があるかを尋ねるようする必要があります。可能であれば、ユーザーがリアルな環境で製品や

プロトタイプを使っているところを見せてもらい、製品の利用方法や利用環境で重視すべきポイントについて知見を得ると良いでしょう。たとえば、製品がもっぱら騒がしい環境で使われていることが分かれば、「騒音環境で使用」という情報をペルソナに追加できます。ペルソナにこのような利用環境に関する情報を盛り込むことができれば、静かな利用環境を踏まえたデザインは最良とは言えないことが事前に分かります。



## UXと文化

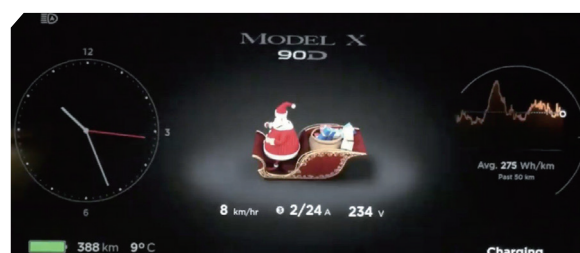
ユーザーの文化の影響は、ローカライゼーション以外の領域にも及びます。ユーザーの動機はたいてい普遍的で性別や文化とは無関係であるにもかかわらず、ユーザーの認知には差異が生じます。事実、文化が異なれば、UXのユーザビリティに対する認知にも違いが生まれます。ある人の文化的な視点を左右する要因には、たとえば社会平等に関する考え方や個性の表出の仕方、不確実性の容認、変化の受け入れ、道楽や贅沢への志向などがあります。100%カスタマイズされたユーザーログインによって個性を尊重するアプリケーションが、ある文化では歓迎され、別の文化では不快感を生むこともあります。文化的なバイアスはユーザーの世界観の基盤となるものであり、従ってユーザーの文化的指向への理解を深めるのは、ユーザーにとって最適なUXをデザインする上で欠かせない取り組みだと言えます。UXデザイナーはこうした文化的指向への理解を、オプションが多いほうがユーザーは喜ぶかどうか、ユーザーを誘導するのとユーザー自身に発見してもらうのではどちらが良いか、インターフェースは最先端のものか伝統的なもののどちらが良いか、といった基本的な方向性を定める際に活用することができます。

## UXと感情

製品開発においては、エモーショナルデザインへの配慮も重要です。ドン・ノーマンが最初に提唱したこの概念は、ユーザーのポジティブな感情を引き出す製品を開発し、それによって優れたエクスペリエンスを生み出すことを目指すものです。ノーマンによれば、ユーザーは感覚に訴えるモノに対して感情的な結びつきを形成し、そこからさらにモノの価値が高まるそうです。エモーショナルデザインへの配慮は、CNCフライス盤や人工呼吸器のような実用的なモノよりも、リモコンのように身体的な使い心地の良さを感じられるモノのほうがずっと簡単かもしれませぬ。しかしながら、感情的な結びつきという

概念への理解を深めれば、どのようなジャンルであっても愛情を注げるモノを作る上で役立つはずで、ここで忘れてはならないのが、感情はモノに触れたときの感覚的経験によってのみ呼び覚まされるわけではないという点です。ユーザーはエクスペリエンス全体を通して、さまざまな感情を覚えます。たとえばゲーム開発の分野では、デザイナーがゲームというエクスペリエンスに明確な段階と前進を盛り込み、物語の起伏を強調するために、プレイヤーが苦しむ、冷静になる、不安を感じる、勝利に酔うといった多様な感情を覚えられよう設計がなされています。これに対し生産性改善アプリケーションなどでは、苦しみや恐れや嫌悪といったネガティブな感情を引き起こす要素は、少なくとも意図的には盛り込まれません。またこの種のゲームを除く多種多様なアプリケーションでは、ポジティブな感情を引き起こすのも定義するのも容易ではありません。産業機械の操作用アプリケーションを使うユーザーにとって「喜び」とはどのようなものでしょうか？機械を操作中に危険が生じたら警告音が鳴ると分かっているならば、安心感を覚えられませんか？医療機器ではどうでしょうか？機器が信頼できる結果を出力してくれると分かっているならば、ユーザーは幸せな気分になるでしょうか？自動車ならどうでしょうか？車載システムと各種デバイスのシームレスな接続が可能なら、ドライバーはワクワクするでしょうか？製品開発においてはこのように、ターゲット市場とターゲットユーザーを知るだけでなく、製品を使った時にユーザーにどう感じてほしいかを見極めることも大切になります。あるアプリケーションにどうしてもなくせない弱点があるなら、弱点を最小限に抑えつつ、その弱点によってユーザーがネガティブな感情を覚えた場合の対策を考える必要があります。たとえば、ユーザーが複雑な変数を入力しなければならないアプリがあったとします。そのようなアプリでは、変数を視覚的に美しく、楽しく表示するなどの方法で、面倒な入力プロセスを乗り越えてくれるユーザーに報いなければなりません。

エモーショナルデザインにはほかにも、ユーザーが製品のエクスペリエンスに慣れてしまうという問題があります。**快樂順応**と呼ばれるこの反応は、製品がユーザーの関心を引かなくなった時に生じます。人はみな新しいものを求めるものであり、製品ユーザーも例外ではありません。製品に慣れたユーザーは使用時に喜びを体験できなくなり、かつては実感できた肯定的なエクスペリエンスもやがて消え失せます。(この反応には良い側面もあります。ネガティブな感情も徐々に消えていくため、否定的なエクスペリエンスもずっとは続きません。)このような快樂順応のプロセスへの対策として、新しい機能やコンテンツを数カ月おきに定期的にリリースし、製品の新鮮さや魅力を維持することが大切です。



テスラは**快樂順応**への対策として、**車載デジタル計器にクリスマスモードを追加した。**

## 初回ユーザーエクスペリエンス

ユーザーと製品の最初のインタラクションは、その後のすべてのインタラクションと異なる重要な要素であるため、FTUE (First Time User Experience 初回ユーザーエクスペリエンス) という略語まで与えられています。FTUEデザインはとても重要です。製品の使い方をユーザーに教え、製品で何ができるか、製品がどう動くかを伝えるものだからです。FTUEデザインが良くなければ、新しいユーザーの関心を引けず、学習曲線は長くなり、ユーザーがその製品の使用を完全にやめてしまうことすらあります。FTUEはユーザーの製品理解の基盤を作るものであり、この初回使用で得た教訓が、やがてユーザーの理解とエクスペリエンスを形成していくのです。FTUEデザインで考慮すべき要素には、以下のようなものがあります。

- **直感性:** 新しいユーザーが見ても直感的に理解できるデザインを目指す。たとえば、自然界や環境を参考にして、物理的なモノと同じように落ちたり、跳ねたりするデザインにすれば、直感的に理解できる。

- **親密性:** ユーザーが利用している既存のUIパターンを参考にして、親密感を覚えられるデザインを目指す。ユーザーのすべてがiPhone利用者と想定されるなら、iPhoneに似た操作やジェスチャーを取り入れたデザインにすれば、新しいユーザーでも気持ち良く使える。
- **一貫性:** 独自の機能やデザインメタファーを取り入れる場合は、製品全体で一貫性を持たせる。そうすれば、いったんUIの動きを学んだユーザーは常に操作の結果を予測でき、操作方法を迷わない。
- **導入障壁の低さ (アプローチビリティ):** 初めてのユーザーでも製品を容易に使い始められるようなデザインを目指す。ログイン/登録/設定のプロセスでやるべきことが少ないほど導入障壁は低くなり、導入初期の学習曲線も短くなる。調査研究によっては導入障壁が高いほうが献身的ユーザーを生むという結果もあるが、その場合、新規ユーザーをつかみにくいというデメリットも生じる。



## UXとフロー

感情に非常に深く関わる概念として「フロー」があります。これは「ゾーンに入る」「リズムにのる」といった言い方もされる概念で、ハンガリー系アメリカ人心理学者のミハイ・チクセントミハイが最初に定義しました。フローとは、没入的な体験の一種で、周りのことを忘れてしまうほどあるタスクに没頭した状態を言います。あるタスクに夢中になり、100%そのタスクに没入した経験は誰しもあるでしょう。たとえばビデオゲームやスキー、ギター、料理などに没頭したことがあるはず。このようなフロー状態に入ると、人は時間の経過や自分自身の肉体、自意識などがまったく気にならなくなり、集中し、充実感に包まれ、対象をコントロールできているという実感を持てます。

「フローとは、何かをすることそのものに完全に没頭した状態である。その時、自意識は消え去り、時間の感覚もなくなる。ちょうどジャズを演奏している時のように、あらゆる行動や動きや思考が、ひとつ前の行動・動き・思考とひと続きになる。フローとは、全身で何かに集中し、持てるスキルを最大限に生かしている状態である」  
ミハイ・チクセントミハイ

このようなフロー状態を生み出すことが、UXデザイナーの究極のゴールです。フローという概念はゲームデザインの世界で広く知られていますが、ゲーム以外のアプリケーションにも十分に応用することができます。UXデザイナーは、ユーザーとその目的の間にアプリが存在していることを意識させないデザインを目指さなければなりません。フロー状態を生むには、タスクの難易度が低すぎず（低すぎればユーザーは退屈します）、高すぎず（高すぎればユーザーはイライラします）、ちょうど良いレベルであることが重要です。ゲームデザイナーがこのゴールを

どのように達成しているかを学ぶことで、より無味乾燥なアプリの開発に携わる私たちも、自社の製品でフロー状態を生み出す方法のヒントを得られるかもしれません。

## エクスペリエンス

ユーザーに関する考察を終えたところで、エクスペリエンスについて見てみましょう。UIは、製品のユーザーエクスペリエンスの大部分を占める要素です。とはいえ、UXイコールUIではない点を忘れてはなりません。エクスペリエンスは、アプリケーションの見た目や製品の外観だけで決まるものではありません。製品開発においては、カスタマージャーニーの最初から最後までユーザーが対象をコントロールしていることを実感できるようなエクスペリエンスを構築しなければなりません。つまり、製品、会社、サービスおよびUIのすべてがUXを形成する要素であり、これらの全要素を組み合わせ、ユーザーとの肯定的な関係構築を目指さなければならないのです。



UXイコールUIではないーエクスペリエンスは、インターフェースだけで決まるものではない。

続くセクションでは、いかなるケースでもUIはユーザーエクスペリエンスの一部に過ぎない、という理解の下、製品デザインにおいてUIとUXに関連するさまざまな側面を考察していきます。

## デフォルトコンフィギュレーション

アプリケーション開発においてソフトウェアエンジニアは、より多くの機能を持たせるべきか、ユーザビリティを考えて簡素化を目指すべきか、絶えず葛藤しています。この課題の解決策としてよく用いられるのが、コンフィギュレーションにオプションを設ける方法です。このアプローチなら、大部分のユーザーはデフォルト設定を利用でき、高度なスキルを持ったユーザーは好みに合わせてコンフィギュレーションを行い、アプリの可能性を広げることができます。ただし、このアプローチでは、大量のオプションを設けるという罫にはまらないための注意が必要です。調査研究でも、デフォルト設定を実際に変更するのはユーザーの5%未満にとどまることが分かっています。従って、ユーザーの使い方を正しく把握してそれに応じたヒントを提供する、ユーザーのスキルレベルを見極めた上で新たなオプションを表示するといった手法を取るほうがずっと現実的でしょう。アドビのPhotoshopおよびIllustratorは、グラフィックデザインとイラストレーションのための極めて高度なプログラムです。しかし機能が多すぎるため、学習曲線の勾配が急になることが避けられません。そこでアドビでは、常時表示されるツールバーの背後に複雑な機能の大部分を隠す、ヒントや裏技は選択時にのみ表示する、プリセットを厳選するといった対策を講じています。

前述した通り、アプリケーションの複雑性はUIだけに原因があるわけではありません。最良のユーザーエクスペリエンスを生み出すには、ユーザーがアプリを実行する前にスキルレベルを見極め、自動的に専門家向け、高度なユーザー向け、初心者向けのコンフィギュレーションを行う必要があります。そのためには、ユーザーが過去に類似分野の同様の製品をどのように利用していたか、自社のシステムにログインしたことがあるか（履歴データがあるか）、アップグレードの履歴はあるか、といった詳細な情報が必要です。このように、製品をユーザーの期待に合わせてカスタマイズ可能にすることで、より幸福なユーザーを生み出すことができます。



デフォルト設定を変更するユーザーが5%未満にとどまることを踏まえ、コンフィギュレーションのオプションをすべて盛り込むのではなく、利用状況に合わせたヒントを提供したり、ユーザーのスキルレベルに合わせてオプションを表示したりするのが望ましい。

## エラーへの対応

ハードウェアの障害も、使えないデータも、不適切な入力もなく、すべてがスムーズに運ぶのなら、アプリケーションのタスクフローデザインはずっと簡単です。しかし現実には、さまざまなエラーが発生します。良いUIデザインと最良のUIデザインの真の違いは、そうしたエラーへの対応方法にあります。エラー対応を、耐えがたいものから許容範囲へと改善するためのヒントをいくつかご紹介しましょう。

- **ユーザー自身がリカバリーできるようにする。**  
 エラーメッセージには、ユーザーができる問題修正方法、またはユーザーには修正方法が一切ない事実を盛り込む必要があります。たとえば、オンラインサービスで接続トラブルが生じたら、その旨を伝えればユーザーは自分でWiFi接続を確認・修正できます。バックエンド側のトラブルなら、その旨を伝えれば、ユーザーがインターネット接続の確認・修正に無駄に時間を費やさずに済みます。
- **ユーザーを知る。**  
 ユーザーがITの専門家なら、「/dev/hd3の検出に失敗しました、RAIDに変えてください」といったメッセージでも十分に事足ります。しかし、ホームセキュリティシステムのユーザーにこのようなメッセージを示せば、ユーザは大きく混乱し、システムへの信頼をなくす恐れがあります。ユーザーの知識レベルとリカバリー能力を把握していれば、エラーの詳細をどの程度伝えるべきか、どのような警告やエラーメッセージを発するべきかが自ずと分かります。またエラーメッセージはユーザーに伝わる用語を使うこと、ユーザーの語彙にある言葉を、ユーザーの知識レベルや専門性に基づいて選ぶことも大切です。
- **ユーザーが自分で修正できないエラーの場合、ユーザーに負担をかけない。**  
 製品に大きな障害が発生した場合、ユーザーが長々とした16進数や専門用語の羅列を読み上げてサポート技師にエラー内容を伝えるようなやり方は得策ではありません。このようなケースでは、技師が遠隔操作で製品を調べられるようにすれば、ユーザーに不便をかけることもなく、また、パフォーマンスを落としてグレースフルフェイルオーバーを行ったのか、無線でソフトウェア更新を行い修正したのかといった判断も下せます。グレースフルフェイルオーバーの一例としては、常時インターネット接続の製品で接続が切れた場合、リアルタイム更新をなしにして実行を継続し、接続が回復したら再同期を行うというケースがあります。ユーザー自身がリカバリーできないエラーの通知をする場合にも、やはりユーザーに余計な負担をかけないことが重要です。ユーザーが自力でエラーを修正できないなら、赤い警告灯で「エラー」の表示を行うにとどめ、ユーザーの役に立たない技術的な通知は省くのが正解です。自動車であれば、「O<sub>2</sub>センサーが正しく作動しません」というメッセージを表示するよりも、赤いエンジン警告灯を点灯させるほうが合理的です。赤い警告灯を見たドライバーはすぐにサービス工場に向かえますが、O<sub>2</sub>センサーが故障したと言われても恐らく何のことか分からず、自分で直すこともできません。

- **ユーザーに主導権を与える。**

ユーザーは自分に主導権がないと感じると、否定的な反応を示します。製品が想定外の状況に陥り、エラーの回避策が複数考えられる場合、最も望ましいのはユーザー自身に回避策を決めてもらう方法です。たとえば、文書内のあるフォントが表示できないなら、そのフォントをダウンロード／インストールするか、品質の劣る既存フォントで代替するかをユーザーに尋ねれば、より望ましいオプションを自分で選べます。ユーザーがこれからやろうとしていることや、かかる時間によって、選ぶオプションが変わってくるかもしれません。

- **情報を紛失しない。**

ユーザー情報の紛失是最悪の事態のひとつであり、ほかに選択肢がない場合を除いて絶対に避けなければなりません。情報紛失は、ユーザーフォームに入力中にエラーが発生した時によく起こります。フォームに入力中、データベースや認証プロセスや接続のトラブルが起こり、トラブル修正後にすべての情報を再入力しなければならないほど、ユーザーにとって腹が立つことはありません。軽視されて気分の良い人などいません。ユーザーが同じ作業を繰り返さずに済むように、適切な対策を講じましょう。ユーザー情報の紛失を避けている好例として、マイクロソフトのOfficeがあります。Officeにはオートリカバリー機能があり、トラブル発生時にユーザーのイライラを上手に軽減しています。アプリケーションが完璧ならクラッシュの心配など不要なので、オートセーブやオートリカバリーといった機能の実装は無駄だと思うかもしれません。しかし現実には、オート

セーブやオートリカバリーでファイルを保存すれば、アプリケーションが情報を紛失する恐れがなく、ユーザーは大いに感謝するものなのです。マイクロソフトがアプリケーションバグからユーザーを保護するためには、甚大な努力が必要だったかもしれません。しかしそうした努力こそが、極めて否定的なエクスペリエンスを、全体として見れば肯定的なエクスペリエンスに変えるのです。

- **タスクフローを中断させない。**

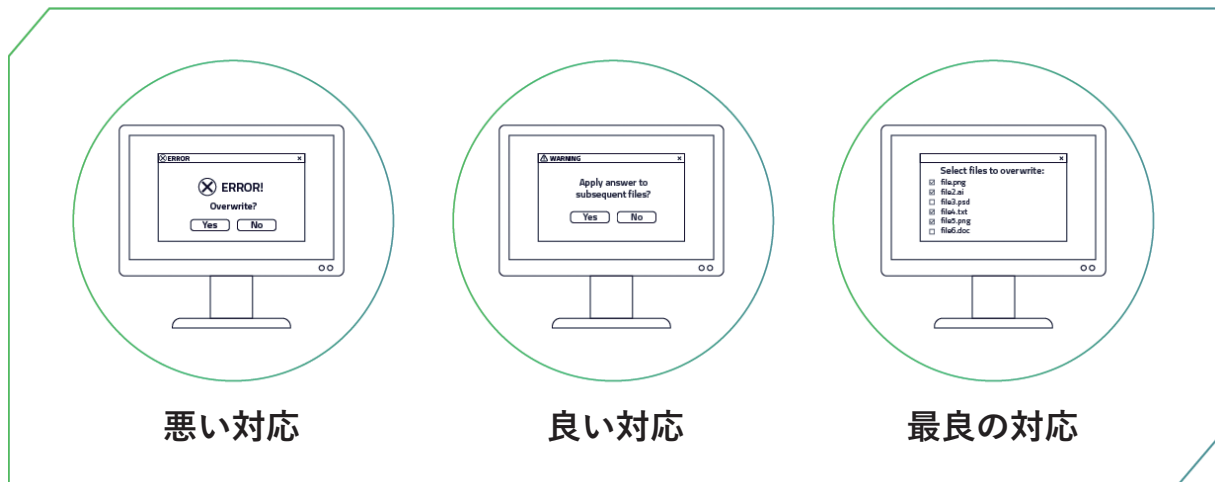
ユーザーが生産的に活動している時は、絶対に必要な場合を除いてタスクフローの中断を避ける必要があります。ユーザーの時間は貴重なものであり、タスクフローを離れたところでエラーに対応できるなら、必ずそうすべきです。また、同じ理由で複数のエラーが発生した場合にはバッチ処理で対応し、ユーザーへの邪魔を最小限に食い止めることも大切です。



*ユーザーが生産的に活動している時は、絶対に必要な場合を除いてタスクフローの中断を避けるべき。*

これらのエラー対応の原則を、いくつかの具体例で見ましょう。

- ユーザーが大量のファイルをコピーしようとしたら、保存先のディレクトリに既存ファイルがあった場合:



悪い対応

良い対応

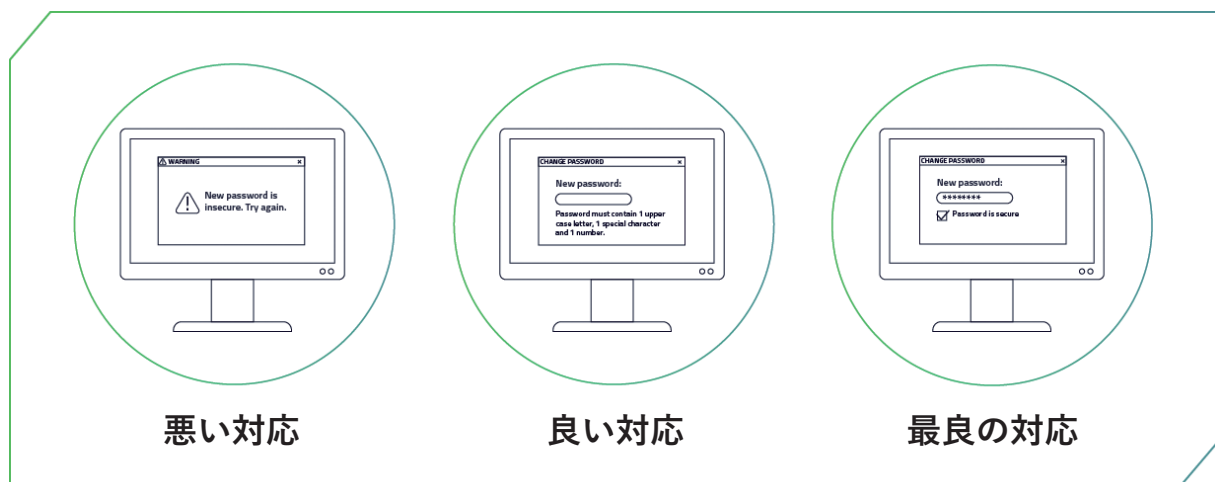
最良の対応

ファイルごとに「上書きする:はい/いいえ」のメッセージを表示してユーザーに都度確認する。

既存ファイルが見つかった最初のファイルで上書きの要不要をユーザーに訊き、残りの重複ファイルも同様に処理するかを確認する。

ファイルの重複の有無を一覧表示し、上書きするファイルにはユーザーにチェックマークを付けてもらい、残りのファイルは別のディレクトリに保存する。

- パスワードの再設定でユーザーに新しいパスワードを作成してもらう場合:



悪い対応

良い対応

最良の対応

新しいパスワードが古いパスワードと同じである、必要な数の特殊文字が含まれていない、長さが足りないといったケースで、「新しいパスワードが安全ではありません。再度お試しください」というメッセージだけ表示する。

事前にパスワード作成ルールを表示し、新しいパスワードにルールに従っていない部分があれば、どこが間違っているのかをユーザーに示し、再度試すよう促す。

事前にパスワード作成ルールを表示し、ユーザーが新しいパスワードを入力している時に緑色のチェックマークで個々のルールに従っているかどうかを示し、ルールに100%従った場合のみ作成フローを進められるようにする。

## インスタントフィードバック UXとファネル

ユーザーのアクションには可能な限り迅速にフィードバックを提供するのがベストです。フィードバックにより、ユーザーは具体的にどのアクションが特定の状況を招いたのかを理解でき、多くの作業を取り消したり、やり直したりせずに、必要に応じてアクションを修正／調整することができます。前述のパスワードの例で言うと、パスワードの強度を示すバーを表示し、強度が上がるごとにバーが長くなったり、赤から緑に色が変わったりする仕組みにすれば、十分に安全なパスワードを作成できているかどうかをユーザー自身が判断できます。さらにこのようなインスタントフィードバックは、即応性のある製品を使っているという実感をユーザーに与え、総合的に見て肯定的なエクスペリエンスを生み出すことにつながります。



学習心理に関する調査研究によると、ユーザーのアクションと関連イベントの間にわずかな遅れがあるとユーザーの無意識下で両者の結びつきが強くなるのが分かっており、具体的にはユーザーのアクションから150～500ミリ秒遅れてフィードバックイベントが起きた時に、結びつきが最も強くなるといわれています。従って、ソフトウェアユーザーに特定の刺激(エラーの回避策表示やユーザーへのリワード提示など)への反応を促したければ、フィードバックの提供をわずかに遅らせるのが、人間の生来の学習回路に合ったアプローチだと考えられます。

製品の購入プロセスもまた、UXの一部です。潜在的ユーザーはまず、ウェブやオンライン、広告、メール、展示会、ソーシャルメディアなどを利用したマーケティング活動を通じて製品を発見します。製品を発見した潜在的ユーザーは、続けて製品の良し悪しを判断します。ここでは、製品がどのように提示され、レビューされ、議論されているかを基にユーザーが製品への期待を形成していきます。より具体的には、営業チームとの商談やダウンロードした製品カタログ、オンラインでの比較調査などが、期待を左右する要素となります。その後、ユーザーが製品を購入して使用を開始すると、購入プロセスで形成された期待が満たされたかどうかの判断が下されます。期待が満たされれば、そのユーザーのUXは良いものだったと言えるでしょう。反対に期待が満たされなければ、製品のその他の要素がどれほど優れていると、ユーザーは失望することになります。

映画の予告動画を見てコメディだと思っていたら、実際にはシリアスなドラマだったなんて経験はありませんか?このようなケースでは、作品の出来がどれほど良くても、期待と実際のミスマッチによる否定的なUXが生じます。ユーザーの失望を防ぐには、マーケティングと製品を一致させなければなりません。マーケティングプロセスで形成されるユーザーの期待と製品使用時の実際の体験とを一致させるには、タスク分析を行い、ユーザーに役立つような特筆すべきポイントに確実に焦点を当てるという方法があります。その上で、マーケティング活動においてアピールしたそれらのポイントを、ユーザーが製品使用時に実際に体験できるようにします。ポータブルスピーカーの広告が「あらゆるモバイルデバイスと連動」とうたっているのに、Android携帯電話でのBluetooth使用が不可能だったら、その製品はユーザーの期待を満たすことはできません。

ユーザーの課題を解決するタスクフローを見極め、それをベースにマーケティング活動を策定しつつ、製品に独自の強みや魅力を織り込んでいくのが最良のアプローチです。製品を利用することで一番の難題を解決できると分かっているならば、ユーザーは使用する前の段階ですでに製品にポジティブな期待を抱いています。そのようなユーザーに、直ちに課題を解決できる合理的かつシンプルな方法(実装やコンフィギュレーション、トラブルシューティングが最小限で済む方法)を提示できれば、ユーザーの中に形成された期待に応えることができるはずです。

## 最も重要なのは「一貫性」

一貫性のあるUXは、良いUXです。製品の利用時に信頼性を求めるのと同じように、ユーザーはエクスペリエンスに対しても常に一貫性があることを求めます。一貫性に対するこのニーズは特に、UXの一部であるUIによく当てはまります。製品のUIが一貫性のあるデザイン言語とデザインメタファーを用いていけば、ユーザーがインターフェースの新しい要素を初めて使う時でも戸惑う心配がありません。ユーザーはアプリケーションについていったん何かを学ぶと、その知識をアプリのあらゆる側面に応用するものです。

ユーザーの大部分は、デザインの各要素やそれらをどう取り扱えばいいのかについて、ある程度の知識を持っているはずです。そうしたユーザーの知識を無駄にせず、製品に有効活用してください。アップルやマイクロソフト、グーグルといった成功を収めている企業は、独自のデザインパターンを一般的なものと普及させてきました。プラットフォームのルールやデザインパターン、ディスプレイのスタイルなどを、一から考えて作る必要はありません。Mac向けのアプリを開発するのであれば、OS Xスタイルのウィジェットやジェスチャー、パラダイムを使うべきであり、同様のことがWindowsやAndroid、iOS向けのアプリにも当てはまります。可能な限りネイティブなデザイン言語を使うのが正解です。

開発した製品が十分に普及すると、クロスプラットフォームで一貫性のある独自のUI/UXを開発したいと思うかもしれません。その一例がネットフリックスです。ネットフリックスはAndroid携帯、スマートTV、Xboxなどいずれのデバイスを使っても、同じエクスペリエンスをもたらされます。ここでもやはり一貫性が最重視されている点に注目してください。複数のプラットフォームで製品を提供するサービスでは、ユーザーエクスペリエンスの違いを最小限に抑える取り組みがなされているのです。さらに、製品としての一貫性のあるクロスプラットフォームUXが良いか、それとも個々のプラットフォームに適合させたUXのどちらが望ましいかは、ユーザーの期待によって異なります。製品を複数のプラットフォームで提供し、ユーザーもさまざまなプラットフォームで製品を利用するのなら、製品としての一貫性のあるクロスプラットフォームアプローチがベストでしょう。反対に、ユーザーが種類のプラットフォームでのみ製品を使うのなら、個々のプラットフォームに合わせたUXのほうが適しています。

またアプリケーション内では同じエレメント、タイポグラフィ、アイコンを使うべきです。視覚的階層を利用したアプリケーションなら、プログラムのメニューツリーにおける現在地やアプリの状態に関わらず、同じ視覚的階層を用いることも大切です。アプリ内でブランディングを変えるのは得策ではなく、他社のサブコンポーネントやサービスを利用する場合でも例外ではありません。

一貫性の重要性はUX全体に関しても当てはまりません。ユーザーと製品開発会社のインタラクションは、ユーザーと製品のインタラクションを通じて一貫していなければなりません。自社の製品、営業チーム、サポートスタッフのすべてが一貫性を維持して、ブランドプロミスを強化し、ユーザーエクスペリエンスをサポートする必要があります。

# UXとKISS

ユーザーエクスペリエンスの大部分には、KISSの原則(Keep It Simple, Stupid、簡潔かつ単純に)が当てはまります。製品開発においては、製品に絶対不可欠なもの、あれば良いものを見極めることが大切です。従って、エンジニアがどれほど強く望もうが、営業チームが懇願しようが、経営陣が求めようが、UXチームと製品管理チームは一致団結してフィーチャークリープ(機能が増えすぎた状態)を避けなければなりません。そのためのひとつの方法が、製品の差別化に寄与する機能に焦点を当てるという方法です。競合他社とターゲットオーディエンスについて深い知識があれば、自社製品を明確かつ簡潔に定義してくれる必須の機能を絞り込んでいくことができます。製品の目的がシンプルであれば、ユーザーは製品の使用時に能率的なエクスペリエンスを実感できます。

## UXプロセスの構築する上で

ここまで読まれた方は、具体的にどうすれば、優れたユーザーエクスペリエンスを自社のワークフローに組み込めるのだろうかとお悩まれるかもしれません。答えは、UXの原則を全社的に浸透させることです。なぜなら、製品のUXを生かすも殺すも、UXデザイナーを中心とした社内のコラボレーション次第だからです。



*UXを生かすも殺すも社内の  
コラボレーション次第である。*

UXデザイナーはユーザーの代弁者となり、ユーザーのニーズや、ユーザーの望む対象へのアプローチ方法についての理解を、他部署もまた深められるよう支援しなければなりません。具体的には、UXデザイナーは以下のような方法で全社的なサポートを提供することが可能です：

- ユーザーストーリーやユーザージャーニーを製品担当マネージャーに提供し、肯定的なUXの創造につながるような製品要件の策定を支援する。
- グラフィックデザイナーと協働してUIのルックを磨き上げ、分かりにくい要素をなくしていく。
- R&Dチームに主要メンバーとして参画し、開発者と協働して、技術的な制約や改善の機会への理解を深めつつ、UIのフィールを磨き上げていく。その際、デザインから実装への移行段階において、デザインの意図が損なわれないよう配慮する。

UXデザイナーがユーザーについて正しい知見を獲得するには、ペルソナの背後に隠された実際のユーザーと時間をかけて慣れ親しむ必要があります。そのためにはまず、その業界とジャンルに関する十分な知識が求められます。そうした知見や知識の一部は競合他社や市場を分析することで得られますが、ユーザーとその課題を真の意味で理解するには、相応の時間を要します。新しい市場に参入したばかりのUXデザイナーなら、ユーザーとそのニーズや動機くらい普通に分かるなどと高を括るのは禁物です。



## デザインの反復

デザインとは段階的に完成していくものであり、UXも含めて、完璧なものはありません。デザインプロセスにおいて反復が非常に重要とされるのもそのためです。何がうまくいき、何がうまくいかないのかを見極めることで、どこを改善し、どこをそのまま際立たせれば良いかが定まってきます。デザインは、先行するデザインと決定を基盤とし、継続的な学習プロセスに沿って構築されていくものです。

UXデザインプロセスに着手するに当たっては、昔ながらのペンと紙を使ったアプローチが初期のアイデア創出段階ではとても大切です。特殊なUXデザインツールと違い、紙は安く、しかも誰でも使えます。紙はまた、新たなコンセプトのスピーディな反復にも適しており、最初からやり直したければ捨てることも可能です。さらに、紙に描くシンプルなスケッチは、基本のアイデアからユーザー入力へと最も素早く移行させる最良のツールでもあります。

紙のデザインはあっという間にその目的を果たしてくれるので、すぐにもっと具体的なかたちでUXアイデアをテストしてみる必要が生じます。その場合はインタラクティブなプロトタイプを作り、実際のエンドユーザーと他のステークホルダーにコンセプトを検証してもらってから、開発フェーズに移るのが最良のアプローチです。またプロトタイプは、できる限り潜在的ユーザーが実際の製品のように感じられるものにするのも重要です。現物を作るのにかかる時間をもたないと思うかもしれませんが、プロトタイプがある程度実際の製品に忠実でなければ、正確な結論を導き出すのは困難です。実物に忠実なプロトタイプには、少なくとも以下の3つの要素が備わっています：

- ビジュアル：スケッチのようにしか見えないプロトタイプではなく、実物に近いプロトタイプであること
- コンテンツ：プロトタイプ内のデータがリアルであること
- 機能：ワークフローが実際の製品のインタラクションを模倣していること

プロトタイプの機能面については、主な画面フローを示した画面を構築し、さらに主だった機能の一部をテストできるような、より詳細なインタラクションを含めるのがお勧めです。プロトタイプに具体的なワークフローが含まれていなければ、その後の改善に必要なフィードバックを得ることはできません。

現在は多数のプロトタイピングツールが市販されており、容易にUXを構築できるばかりか、そのうちの多くは実際の製品に簡単に応用することもできます。インタラクティブなプロトタイプは、主なユーザーニーに照らし合わせてデザインを検証し、ユーザーの弱点(と喜びの瞬間)を見つけ、ユーザーナビゲーションフローへの理解を深めるのに役立ちます。またプロトタイプは、製品UI/UXのアイデアを共有するツールとしても有益です。インタラクティブなプロトタイプを使えば、デザインの意図を開発者やマネジャー、顧客に伝えることができます。

デザインの反復と開発は同時進行させて、プロセスのスピードアップを図ることもあります。これはアジャイル開発に適した手法です。このような並列アプローチでは、デザインの反復に着手してから実装フェーズに入るのが普通で、また、一度にひとつのユーザーストーリーに焦点を当てるのが一般的です。

最初のプロトタイプがうまくいったら、プロトタイプのテストと改善を進めながら、開発に着手します。また、UXデザイン／プロトタイプ／テストのループにおいて明らかになった修正点は、すべて開発スクラムにフィードバックします。これらのフィードバックにより、デザイン業務の合理性を維持し、無駄な開発作業を最小限に抑えられるほか、並列アプローチで進行することによって開発に着手するまでの時間を最短化することもできます。さらに、デザインを開発ループにしっかりと織り込むことで、構築済みのデザインを後から廃棄するような無駄も最小限に食い止められます。

## 経時的な進化

プロトタイプはまた、アプリケーションの使い方が経時的にどう変化するかを浮かび上がらせるものでなければなりません。ユーザーの製品の使い方は最初の使用時から進化し、気まぐれに使ったり、定期的に使ったりと変化していきます。このようなユーザーのニーズの変化に伴い、製品も変化する必要があります。購入から一年後にも製品使用時の喜びを感じてもらうには、新規ユーザーと使い慣れたユーザーの両方の使用パターンを考えたプランニングが求められます。たとえば、メールソフトの新規ユーザーはアカウントを設定した後、さほどスクロール操作をせずに少数のメールを管理することができます。しかし一年後には、このユーザーは数千通のメールを検索したり、既読と未読に分けたり、容易にメールを見つけられるように優先順位を付けたりする必要が出てきます。このようなソフトウェアのテストでは、メールが10通の場合と10,000通の場合の両方を想定し、いずれの場合にも肯定的なエクスペリエンスをもたらせるかどうか検証しなくてはなりません。

## UXテスト

では、ユーザーエクスペリエンスのテストはどのように行えばいいのか？テストを実施するには、「製品、ユーザー、メトリクス」という3つの要素が不可欠です。製品は開発サイクルの段階に応じて、プロトタイプでも実際のハードウェアでも構いません。ただしプロトタイプの場合は、色やフォント、反応性、触感といった要素を用いて、できるだけ実際のUXデザインに近づける必要があります。ユーザーによるテストが必須なのは、製品の良い部分を洗い出し、悪い部分を取り除き、製品が使いやすく、望ましいものに仕上がっているかどうかを判断するためです。最後のパフォーマンスメトリクスについては、判断の基準として必要なものであり、ベースラインと比較してUXの改善が成功したかどうかを評価します。メトリクスの例には以下のようなものがあります：

- 機能面のクオリティ：タスクにかかる合計時間、必要なクリック回数、タスクの達成率、エラー発生数など、ユーザーの行うタスクの機能面のクオリティ。
- 感情的な側面：製品の使用中にユーザーが体験する喜びやいらだち、退屈などの感情的な側面。
- ページビュー、ダウンロード、インストール回数：ユーザーエクスペリエンスの最初の部分をトラッキングすることで、ユーザーがファネルの初期段階から製品のインストールに十分な関心を示せるかどうかを判断する。
- 評価(星の数)、レビュー、カスタマーフィードバック：ユーザーが自由意思で製品について高く／低く評価した内容。

UXテストは、定性的にも定量的にも行えます。定性的テストでは、小規模なフォーカスグループから世界中のオーディエンスを対象にしたオンラインアンケートまで、幅広い方法でユーザーのニーズと動機について知見を集めます。定性的テストでは、ユーザビリティを検証したり、ユーザー調査を行ったり、ユーザーから直接フィードバックを収集したりします。これらの定性的テストで得られたユーザーアンケートやユースケーステスト、動画撮影、観察レポートなどの結果を用いることで、デザインの問題点を明らかにし、ユーザーエクスペリエンスに関する知見を獲得することが可能です。一方、定量的テストはUXテストの数値的な側面を担います。ユーザーの行動を通して、デザインのユーザビリティを間接的に評価する手法です。定量的テストで得られるログやイベント、タイミング、回数といったデータからユーザーの動機を理解するのは難しいかもしれませんが、ユーザーがどのように製品を使うかを知る上では有益です。またユーザーの思考プロセスを邪魔せずに測定値を収集できるため、定量的データは定性的データを集めるよりずっと容易なケースが多々あります。

## まとめ

本ホワイトペーパーを通読された方にはきっと、アマゾンやアップル、コストコ、ディズニーといった業界リーダー各社がUXに大規模な投資を行うのも当然だとご納得いただけることと思います。これらの企業はUXに投資することで、より良い製品を生み出し、開発コストを削減し、顧客に今まで以上の幸せをもたらし、平均を上回る収益成長を遂げ、競争優位性を高めているのです。[アドビの依頼により米国の調査会社フォレスター社が実施した調査](#)によれば、ユーザーエクスペリエンスを重視する企業はそうではない企業よりもブランド認知度が1.6倍、平均注文額が1.9倍、顧客維持率が1.7倍、投資利益率が1.9倍、顧客満足度が1.6倍高いそうです。

ただし、定量的データはユーザーに関する「なぜ」への答えは教えてくれません。従って定量的データは、UXのパフォーマンスを基準と比較する、経時的なユーザビリティの質をトラッキングする、ユーザーの行動に関する統計データを集めるといった目的で利用するにとどめるべきでしょう。定量的データからUXに持続的な問題があることが分かったら、ユーザー調査で得た定性的データを組み合わせて分析すれば、その問題の根本原因を発見することができます。

「エンジニアによるデザインの多くが抱える問題、それは論理的過ぎるという点だ。エンジニアは人の行動をありのままに受け止めるべきで、こうであってほしいという論理で解釈してはならない」

—ドン・ノーマン

現状、貴社が競合他社に比べてユーザーのニーズにあまり注意を払っていないのであれば、その影響は直ちにビジネスに表れてくるはずです。

製品のUXに投資するには確かに時間がかかるかもしれませんが、しかしながら、UXへの投資を始めるなら今しかありません。エクスペリエンスを重視した製品の開発をご検討中なら、The Qt Companyにぜひお問い合わせください。

## 著者について

アンティ・アールトネン (Dr. Antti Aaltonen)

アンティ・アールトネンはThe Qt Companyのユーザーエクスペリエンス統括者として、UI/UXデザイナー向けのツールデザインチームを率いています。デザインとテクノロジーと製品開発という3つの領域を横断した職位を愛するアールトネンは、使い勝手に優れ、使う喜びにあふれ、洗練されたエクスペリエンスを備えた製品の開発に20年にわたり携わってきました。The Qt Companyに入社する以前は、ゲームシリーズ「アングリーバード」の開発で知られるロビオ社 (Rovio) で7年間にわたりUXチームを統括。その前にはノキアに10年以上勤務し、ノキaN9 UXチームの創設メンバーのひとりとして活躍しました。2007年にオランダ・フローニンゲン大学でコンピューターサイエンス博士号を取得しています。



The Qt Company develops and delivers the Qt development framework under commercial and open source licenses. We enable the reuse of software code across all operating systems, platforms and screen types, from desktops and embedded systems to wearables and mobile devices. Qt is used by approximately one million developers worldwide and is the platform of choice for in-vehicle digital cockpits, automation systems, medical devices, Digital TV/STB and other business critical applications in 70+ industries. With more than 250 employees worldwide, the company is headquartered in Espoo, Finland and is listed on Nasdaq Helsinki Stock Exchange. To learn more visit <http://qt.io>

© All rights reserved