

Web デザインの終焉

ヤコブ・ニールセン

2000 年 7 月 22 日

Web サイトは、あらゆる点で、**個性的な外観と独特のデザインを控えめにしなければなら**ない。

- ・ ビジュアル・デザイン
- ・ 専門用語とラベリング
- ・ インタラクション・デザインとワークフロー
- ・ 情報アーキテクチャ

これらの変化は、すべて同じ結論に繋がる 4 つの異なるトレンドによって引き起こされる。

1. インターネットの UX に関するヤコブの法則

ユーザは、殆どの時間を他のサイトに費やす。つまり、ユーザは、あなたのサイトが、既に知っている他のすべてのサイトと同じように機能することを好む(ヤコブの法則について更に詳しく説明している [2 分間のビデオ](#)をご覧ください)。

この法則は、数年来 Web を支配しているので、将来のトレンドでさえない。Web サイトのデザインが標準化されればされるほど、より多くのビジネスを呼び込むというのは、長い間、真実だった。Yahoo と Amazon を考えてみればいい。”ショッピング・カート”と莫迦げた小さなアイコンを考えてみればいい。青色のテキスト・リンクを考えてみればいい。

2. モバイル・インターネット

モバイルは小さな画面(簡単に持ち運べる唯一のものだから)へと突き進み、たいていの場合グレースケール(バッテリーを節約するため)になるだろう。モバイル帯域幅は、有線帯域幅よりもはるかに制限されるだろう。[私は現世代の WAP 電話を信じていない](#)が、もっといいデバイスが手に入れば、モバイル・インターネットは大きくなると確信している — それでも、これらの次世代モバイルは PC よりも小さい画面になるだろう。こうして、コンテンツとソリューションに重点が置かれる。最も必要なもの以外、**ナビゲーション機能に画面スペースを割いてはならない**。ナビゲーション用のスペースが少なくなると、どこへ行き、どのようにオプションを説明するか、について標準的な習慣に従うことがより重要になっていく。

3. ネットワーク・コンピューティング

[ネットワークとは](#)、場所やデバイスを越えてコンピュータや情報機器の活用を統合し、全体をシームレスに構成する UX である。別のデバイスを使用するたびにルールが変更されたら、どうやってシームレスになると言うのか? モバイル・インターネットを簡素化する必要があるなら、有線インターネットもそうすべきだ。

複数のデバイスにわたってサービスを提供する場合、ユーザはそれが同じサービスであることを認識できる必要がある。同じ機能の多くは、各プラットフォームで提供した方がよいが、意味がないとか、適切なユーザビリティを提供するのが難しいとかいうデバイスの場合、一部の機能は省略されたり、デバイスの裏側に追いやられたりするかもしれない。こういうふうにと考えると、表現ではなく意味論に重点を置かざるを得ない。

4.同時配信されたコンテンツとサービス

統一化した Web サイトの時代は過ぎ去った。1993~1998 年頃にかけて、殆どの Web サイトは古代ローマ軍のキャンプのようなものだった。バリケード内のすべては、キャンプの住民によって慎重に計画、構築されていた。(防火)壁は支配の終わりを示していた。外部はすべて荒野であり、サイトとは接続されていなかった。

これは、1991~1992 年頃の Web 黎明期とは対照的だ。この時代、特定のサーバ上のコンテンツは、Web のおこぼれと接続されているよりも、サーバ上のどこにも接続されていない方がはるかに多かった。Web は一体であり、1つのサイトに属するページが特別に扱われることはなかった。

1998 年頃からは、サイトの内外を流れる同時配信コンテンツに Web サイトが依存することがより一般的になった。複数の Web サイトに表示される可能性のあるコンテンツを作成する場合、コンテンツ・デザインを、見出し、箇条書きリスト、強調表示されたキーワードなど、どこでも通用するいくつかのメカニズムに制限することが必要になってくる。

同様に、Web サイトが、その機能とコンテンツの多くをインポートする場合、通常、リソースを節約することとか、インポートを簡素化してサイト内に収まるように、できるだけ労力をかけないこととかが必要になる。すべてがほぼ同じである限り、上手くいく。あまりにも特別すぎると、衝突が発生する。

また、**アプリケーション・サービス・プロバイダー**は、Web サイトが過度に個性的なデザインを維持することを難しくしている。Web サイトの機能の一部が、メーリング・リスト、検索、会議登録、ショッピング・カート、プロモーション、クーポンといった特定の専門的機能を提供する他のサイトに存在することが一般的になりつつある。ユーザが外部委託された機能を使用する場合、特定の機能を使っている間、一時的に別のサイトに移動してしまったことに気が付かないことが理想的である。その感覚は、**単一のスムーズなやり取りの中に留まる**ようなものでなければいけない。

現在、ASP はクライアント・サイトの表示に近づけるため、ページを編集する手段を限定的に提供しているが、外部委託されたページを、ローカルでホストしているページと同じように感じさせることは、どちらも非常に単純なデザインでない限り、通常は難しい。

専門的ユーザのサポートは、制御を放棄することをサイトに求める

過去 5 年間、Web は初心者ユーザに重点を置いてきた。基本的に、すべての Web ユーザは常に初心者である。なぜなら、専門的ユーザになれるほど、十分に長く個々の Web サイトを利用することは滅多にないからだ。

[熟練したユーザ](#)になれるほど頻繁に訪れるユーザがいたところで、数秒で使用方法がすぐに分からないようであれば、ユーザは Web サイトにアクセスしないため、サイトは初心者をターゲットにしたデザインにする必要がある。**学習時間 0、さもなくば廃れるのだ。**

高度な機能に対する経験豊富なユーザのニーズと、極度の単純さに対する初めての訪問者のニーズとの間の緊張を解消する方法は、専門的機能をブラウザや他のクライアント・ソフトウェアに移行することである。“戻る”ボタンとブックマークは、2つの簡単な例だ。サイトのドメインから除外されているため、どちらも上手く機能するし、どこでも同じように機能する(愚かにも標準を無視したサイトを除く)。

専門的機能がすべてのサイトで標準化されているか、クライアント・ソフトウェアでサポートされていれば、経験豊富なユーザは、サイト・デザインで視覚的に表示する必要なく、その機能を利用できる。従って、初心者ユーザにとって学習上の問題は発生しない。それどころか、他のサイトで[学習したことを真似する](#)ことができるため、そのサイトで学習する必要なしに、**サイトへ初めて訪問するユーザは専門的機能を使用することができる。**

Web デザインに残るもの

Web サイトがより似たものになり、外観デザインがより単純化されたとしても、個々のサイトの使いやすさを最適化するために、依然として多くのデザイン上の意思決定を行う必要がある。

最も重要なことは、各インターネット・サービスは、特定のユーザとそのニーズの**タスク分析**に基づいている必要があるということだ。あなたは、標準化されたユーザ・インターフェイス要素を様々な方法で組み合わせることができるし、より優れたサイトが、ユーザが問題に取り組みたい方法をサポートしてくれるだろう。

例えば、常に検索を“検索”と呼び、単純な検索と高度な検索をいつでも同じ方法で区別している場合でも、高度な検索が特定のサイトにとって意味があるのかという問題は残るだろう。

[コンテンツ・デザイン](#)も残るだろう。商品説明はそれぞれ異なる。それぞれの意見は異なる。情報の各ユニットを記述する最善のアプローチを決定する必要が常にあるだろう。[情報アーキテクチャ](#)は部分的に標準化されていくだろう。すでに起きている例としては、殆どの企業 Web サイトの“会社案内”のエリアがある。すべてのユーザは、このエリアに、経営陣、社史、財務情報と投資家情報、PR とプレス・リリース、及び採用情報に関する

サブサイトが含まれていることを期待する。しかし、これらのサブサイトの構造は、特定の企業の特徴によって異なる場合があるかもしれない。同様に、個々の製品やサービスに関連して、サイトごとに構造が異なる他の多くのエリアが存在することになるだろう。