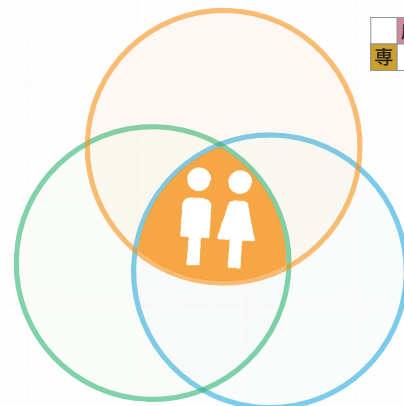


4.

システム開発の効率化と デザイン品質の向上

—テンプレート活用による HCD の効果的な実践—



善方 日出夫 (富士通デザイン ソフトウェア&サービスデザイン事業部 フィールド・ソリューションデザイン部)

システム開発における HCD プロセス 実践の課題

近年、コンシューマ向けのアプリケーションでは当たり前になっているユーザビリティの高さが、BtoBを対象とした業務システムにも求められてきている。業務系アプリケーションは、利用者や利用範囲が限定的であるなど、どうしても機能が優先され、ユーザにとっての使いやすさにはあまり配慮されてこなかったように思われる。しかし、それによるヒューマンエラーの誘発や手戻りによる開発コスト増加の問題などが知られるようになり、たとえば2009年に電子申請システムを対象とした電子政府ユーザビリティガイドライン¹⁾が公開されるなど、ユーザの視点に立った開発の重要性が広く認識されてきている。このような、ユーザビリティを向上するための方法論としてHCD (Human Centered Design) プロセスがあるが、ものづくりとしての開発プロセス全体を通じた中での実践が必要になってくる。

SI構築を中心とするシステム開発を行うにあたり、富士通にはSDEM (Solution-oriented system Development Engineering Methodology) という標準プロセス体系がある。ここでは、作業の漏れを防ぎかつ効率的にプロジェクト運営ができることを主な目的として、一般的な開発プロセスと同様、企画・設計・開発・テスト・運用保守といった工程での必要な作業項目や成果物が広く体系的に記述されてい

る。ユーザビリティ向上を目的とするHCDプロセスも、2007年よりこのSDEMの中に組み入れられている。

そこでのユーザビリティ向上活動として、具体的には、企画工程における「①情報システム戦略への組み込み」、 「②ユーザビリティ要件の明確化」、要件定義工程における「③利用状況の把握と利用者特性および環境の記録」、開発工程における「④ユーザビリティの設計・開発」、運用テスト工程における「⑤ユーザビリティの評価」「⑥システムの導入と運用・支援活動」、プロセス全体を通じた「⑦人間中心設計のプロセスの計画と実施」といった内容が整理されている (図-1)。

このようなSDEMの標準プロセスにのっとったかたちでHCDプロセスを実践していくことが本来の理想であるが、実際の開発現場においては、HCDの重要性は理解されているものの (知られていない場合もいまだにあるが)、そのための予算がなかったり、必要な専門家のリソースが足りなかったりといった現実的な課題もある。また、そのような開発案件を支援する役割を持つデザイン部門でも、HCDプロセス実施のためのノウハウはあるが、製品数や案件数に応じきだけのリソースや時間も不足しているといった課題がある。

このような課題の解決のためには、開発の現場にいるSEや開発者自身が、ユーザビリティに関する知識やノウハウを身に付けることが長い目で見れば最短であり、そのための教育コースなども準備され

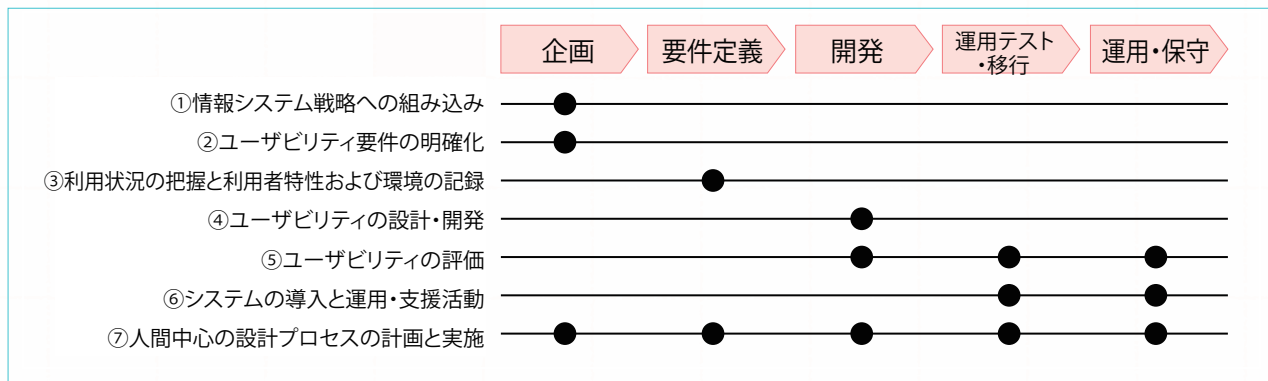


図-1 SDEMにおけるユーザビリティ向上活動

ている。一方で、そのような新たな工数を確保できない多忙な現場も多くあり、そのような場合は即戦力としてのノウハウのツール化が有効な手段になってくる。HCDプロセスの上流は難しいが、「④ユーザビリティの設計・開発」や「⑤ユーザビリティの評価」のフェーズでは、たとえば、画面テンプレートやチェックリストといったある程度の型決めをしておくことで、デザイナーやHCD専門家がかわらなくとも、SEや開発者自らがその型を使うことで、一定レベルのユーザビリティを確保することができるようになる。またそれは、すでに型が用意されているという意味で、より効率的なシステム開発につながっていく。

本稿では、ユーザビリティを高めるためのテンプレートがシステム開発効率やデザイン品質に与える影響について、富士通デザインで開発した「富士通GUIデザインプラットフォーム」の事例をもとに、テンプレート制作の背景や特長とともに紹介する。

どのようなユーザビリティの型決めが可能なのか？

業務アプリケーションを中心とした、ユーザビリティ評価や改善設計等の業務を数多く実施していくと、扱う対象に関する共通性が見えてくる。それは、「ユーザが行うタスクを画面という切り口から見た場合、どのシステムでも似たような画面構成になっており、典型的で単純な画面種類と画面遷移に落とし込むことができる」というものである。

この場合、ユーザインタフェース上の特徴から見

ると、文書作成などを主目的とするオフィス系のソフトウェアや、Visual StudioやPhotoshopなどの開発・制作系のソフトウェアではなく、データベースと連携し主にWebブラウザ上で実行されるような業務システムがその対象となる。

前者は、ドキュメントやプログラムを制作することが目的であり、真っ白なキャンバス上でテキストや画像などを扱うことが中心となる。したがって、1つの作業ウィンドウが中心的な役割を果たすため、画面数や画面遷移は少ない。一方、後者は、データベースの情報を新たに追加したり内容を編集したりというやりとりが中心となるため、機能に応じた画面の種類や遷移が発生する。

なお、ここでの画面種類と画面遷移は、ソフトウェア品質の観点から見ると、利用品質に影響を与えるシステムのユーザインタフェース、つまりソフトウェアの内部／外部品質であり、このレベルでユーザビリティを捉えることで、具体的な画面としての型決めが可能となる²⁾。

部品レベルでのUIの型決めということも有効であり、UIパターンとしてツール化している事例もあるが³⁾、利用品質に影響を与える単位としては、部品レベルのパターンの組合せから構成される「画面」の方がより重要であり、その画面単位での型決めを行った。

基本 9 画面を定義しテンプレートに落としこむ

ユーザの行うタスクを操作フローとして時間順に

見ていくと (図-2), ユーザはまずそのシステムを使うためにログインする (①ログイン画面). ログインした後は, 大抵の場合, メニューを選択する画面に遷移する (②メニュー画面). メニュー画面上では行うべき業務 (機能) を選択することになるが, データベースとやりとりするシステムの場合は, まずメニュー選択後は, 各機能のトップにおいて検索機能が用意されていることがほとんどである (③検索条件画面). そこで検索を実行し, ユーザが処理すべき対象を一覧する (④検索結果一覧画面). この際に, 同一画面上で検索と結果の一覧表示をする場合もある (⑤検索条件+結果一覧画面). その中から, 対象となる 1 件を特定し, それに対して編集や更新, 削除といった処理を実行する (⑥入力画面). また, 新規にデータを作成する場合も同じ画面で対応できる. そして, 新規に入力した後, あるいはデータを更新した後は, その内容を確認し (⑦詳細画面), 一連の作業を完了する.

以上を, 最も単純化した際の画面種類と画面遷移の典型と捉えた. 処理の側面からは, 新しくデータを作る場合 (new) とすでにあるデータを対象に処理する場合 (edit) の 2 つの視点で整理されているかどうか, またそれが分かりやすくなっているかどうか, ユーザビリティ上のポイントとなる. また, ユーザの意図の側面からは, 「探す」「作る」「見る」の 3 つに大きく分けることができ, それぞれ, 「検索」「新規作成」「参照」などの機能に対応するといえる.

これらを意識した画面の種類とその遷移を, 提供する機能とともに業務システム全体で整理することがユーザビリティ向上にとっての基本的なポイントとなる.

画面のテンプレートという観点からは, 上記の 7 画面に加えて, Web ブラウザ上での文章表記のサンプルとするための⑧テキスト画面, 日付入力など入力時の補助的畫面としての⑨入力補助画面の 2 つを加えて, Web 業務アプリケーションの基本 9 画面として定義した (表-1).

現場ですぐに使えるツールの提供

基本 9 画面が定義できたことにより, これらのテンプレート化を実施した (図-3). テンプレート作成にあたっては, 画面レイアウトの原則を踏まえたヘッダやフッタのエリア分けとそこでの役割の定義, 各画面機能を実現するためのコントロール類やボタンの配置などを最適化, 標準化している. また, 色などのデザインテイストに関するバリエーションも, 多様な利用シーンに対応するために約 50 テーマの中から選択できるようにし, アクセシビリティに関しても事前に配慮し文字の視認性なども最適となるよう調整している. また, Web 標準にのっとった HTML と CSS により制作しているため, 簡単かつ効率的にカスタマイズできるようになっている.

さらに, ツールとしては, テンプレートとともに各画面のユーザビリティ上のポイントを記したガイドラインも併せて準備している. 画面単位でのユーザビリティのポイントはいくつかの視点があるが, たとえば, メニュー画面では, 「ログイン ID やパスワードの入力フィールドとログインボタンは近い位置に配置す

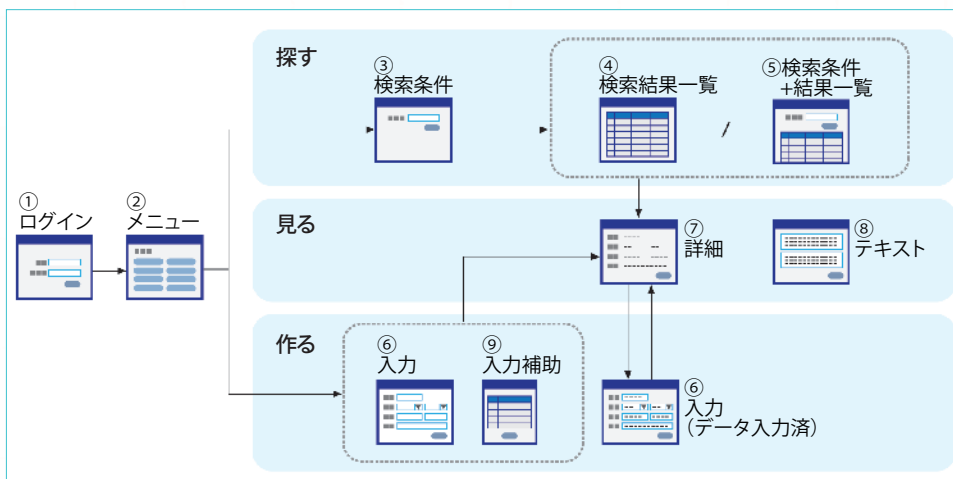


図-2 基本的な画面と操作の流れ

画面の種類	役割
①ログイン	ユーザID、パスワードを入力し、システムにログインする。
②メニュー	機能をメニュー項目として整理・表示し、作業の入り口を提供する。
③検索条件	検索条件を入力し、検索を実行する。
④検索結果一覧	複数の検索結果を一覧表示する。その中から対象を特定し、次処理（詳細表示、あるいは、入力）へ遷移する。一覧からの選択に関しては、単一か複数か、実行に関しては、その選択肢が単一か複数かにより組合せがある。
⑤検索条件 + 結果一覧	検索条件の入力、結果一覧の表示を一画面で行うもの。繰り返して検索することが多い場合などに用いられる。
⑥入力	必要情報を入力し、処理を実行するフォーム画面。入力内容が多い場合や手順を踏む場合などは、複数画面（ページ）にまたがることも多い。
⑦詳細	詳細をすでに入力済み（あるいはDB登録済み）のデータの詳細を表示する画面。
⑧テキスト	一定程度のまとまりを持った文章や段落からなり、お知らせ事項やヘルプなどを表示する。
⑨入力補助	主となる入力操作の流れの中で、入力を補助する機能として働く画面。たとえばカレンダー、住所検索など。ポップアップ的なUIであることが多い。

表-1 基本9画面とその役割

る」とか、入力画面では、「必須項目が必ず分かるよう明示する」といった基本原則的なものから、一覧画面からの項目の選択と実行、たとえば、編集や削除をする場合、一覧項目の一つひとつにそれぞれのボタンを設けるのか、あるいはまず項目を選択しその後1つのボタンでそれを実行するのかといった、やや複雑なものも記述している。これらのポイントは最後にまとめてチェックリスト化し、アプリケーションの設計や開発時のユーザビリティの確認と品質確保のツールとしても使えるようになっている。

このように、SEや開発者は、利用目的に応じてそのままテンプレートをカスタマイズするだけでなく、その際、ガイドラインを参照しながら画面レイアウトの変更や色彩の調整などのカスタマイズを必要最小限にとどめることで、一定レベルのユーザビリティを確保した画面の設計／開発を行うことができる。また、テンプレートを利用することで、画面をゼロから作る必要がないため、開発効率の向上にも期待ができる。

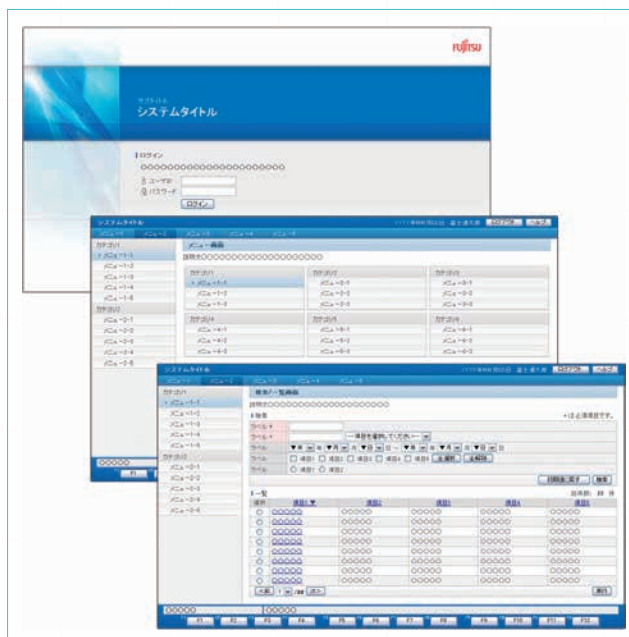


図-3 テンプレートのサンプル

利用の実際と効果

富士通デザインでは、このようなツールを2007年から「富士通GUIデザインプラットフォーム」として、富士通グループに社内提供している。テンプレートの種類は、まずはHTMLからスタートし、開発現場のニーズや要望を取り入れながら、ASP.NETや富士通の開発プラットフォームに応じたものなどを順次用意してきた。

開発者はイントラネットのダウンロードサイトから、必要なテンプレートやそれに付随するガイドラインを一式ダウンロードして使うことができるようになってきている。現在でも年間で約2,000本前後のテンプレートがダウンロードされており、自部署での簡易なツール開発から、お客様商談での活用まで、幅広い範囲で継続的に利用されている。

これまでのアンケートやヒアリングなどから得られたツールのメリットを以下にまとめる。

全般的効果

- スタート時に色合いやレイアウトをメンバー間で共有できるため、方向性のズレなく開発が進められる。
- 文字や色合いのバランスを考える時間が短縮できる。
- 本物に近い画面イメージを提案でき、具体性という点でアピールできる。
- Web画面のひな型という発想はとてもユニーク

で、ツールとしてよくできている。

デザインやユーザビリティ品質に関する効果

- 開発者がデザインについて考えなくてよい点が評価できる。
- 文字や色合いのバランスがよい画面が簡単に作成できる。
- ボタンの配置など統一でき、ユーザビリティが向上できる。

当初このツール（テンプレート）の提供は、実際のシステム開発の中で利用してもらうことを前提としていたが、開発に入る前の提案フェーズでも活用されていることが分かってきた。これは、事前にテンプレートを利用してモックアップを作成することで、提案書の中での画面イメージや、お客様へのプレゼンテーション時のデモサンプルに使うというものであり、そうすることにより提案段階でお客様により具体的なシステムイメージを持っていただくことができるようになった。単に画面機能としての標準化にとどまらず、デザイナーによる審美性の高いテンプレートを提供していることも、このような効果的な使われ方につながっている。

また、画面の制作効率に関して比較実験も行い、HTML/CSS の初心者が模擬の画面制作を行った結果、テンプレートを利用することによって、利用しなかったときよりも開発工数が約 4 割程度削減されるという結果を得ている。まったくのゼロから作るよりも効率がよいのは当然としても、大きな工数削減効果があるといえる。

このように、テンプレートを活用することで、システム開発の効率化やユーザビリティ品質の向上がはかれるとともに、提案段階やシステム開発における早い段階からの利用イメージの伝達に有効に機能していることが分かる。これは、ユーザの視点を積極的に取り入れる HCD 開発が、実際のものづくりの中だけでなく、SI ビジネスでの優位性獲得においても重要な要素になっていることを示している。

より開発プロセスで使える HCD のために

SI 開発における HCD プロセスの実践として、プロセス後半の設計や評価で使えるユーザビリティの型決め／標準化の事例を紹介した。

このような取り組みは、すでに他社でも行われており、HCD の実践活動における同様の課題感やその解決策が見出されている。システム開発のフェーズごとの画面デザインパターンのあり方について、より研究的な側面からアプローチし、その特徴と体系について論じたものもあり、外部設計での画面デザインパターンの有効性や、企画・提案準備フェーズでの現場ニーズについて、同様の報告がある⁴⁾。

今後は、プロトタイプングツールとの連携や、HTML5 による実装を前提としたテンプレート、スマートデバイスに対応する新たな画面パターンの導出といった、新しい技術やデバイスへの対応を検討していく必要がある。また、非機能要求グレード⁵⁾で用いられているグレード表のようなカテゴリ分類を設定し、それに対応付けた画面パターンや遷移パターンが整理できると、上流プロセスに応じた形での画面デザインの型決めが自動化でき、開発現場ではより有効なツールとして受け入れられてくると考えられる。

参考文献

- 1) 内閣府：電子政府ユーザビリティガイドライン，2009，http://www.kantei.go.jp/jp/singi/it2/guide/security/kaisai_h21/dai37/h210701gl.pdf
- 2) 永野行記，宇多村志伸，善方日出夫：情報システム開発におけるユーザビリティ要件，日本人間工学会第 53 回大会講演集，p.176 (2012)。
- 3) ソシオメディア UI デザインパターン，<https://www.sociomedia.co.jp/category/uidesignpatterns/>
- 4) 大久保亮介，野田尚志，谷川由紀子，福住伸一：HI 設計におけるデザインパターン開発，NEC 技報，Vol.64, No.2 (2011)。
- 5) (独) 情報処理推進機構 ソフトウェア・エンジニアリング・センター：非機能要求グレード (2010)，<http://sec.ipa.go.jp/reports/20100416.html>

(2012 年 8 月 22 日受付)

善方 日出夫 ■ zempo@jp.fujitsu.com

1995 年上智大学大学院文学研究科博士後期課程修了。1998 年富士通 (株) 入社。認定人間工学専門家，HCD-Net 人間中心設計専門家。