

## 91

# 利用時品質を高めるための開発プロセス ～デザインエンジニアリング～<sup>1</sup>

## 1. 概要

本稿では、車載器向け HMI (Human-Machine-Interface) のプロトタイプ開発において、ツールを活用することで実現したエスディーテック株式会社 (以下、同社とする) のデザインエンジニアリングの適用事例を紹介する。活用したツールの開発経緯やその適用効果についても報告する。

同社は 2015 年の設立時より「利用時品質の高いモノをつくる」ためのプロセスとしてデザインエンジニアリングを提唱し、開発プロセス改革の提案および独自開発ツール群の提供に取り組んできた。

特に HMI 開発ツールである「TRITO® Linkage」ではデザイン工程とエンジニアリング工程の融合を積極的にうながし、従来は後回しにされることが多かった利用時品質の向上を開発段階で図ることを可能としている。

## 2. 取組みの目的

近年、User Experience (以下 UX) という言葉が様々な製品開発において重要視されるようになってきている。自動車においてもこれは同様で、特にコネクテッドカーや自動運転といった大きな技術的な節目を迎えている現状において、従来の「走る、曲がる、止まる」という自動車の根幹を構成する機能と同等またはそれ以上に、UX が果たす役割の重要性が認識されてきているといえるだろう。中でも自動車とユーザの接点となる HMI の果たす役割とその重要性は年々増してきている。一方で、UX の向上という点に関しては、なかなか決定的な手法を確立するには至っていないというのが現実である。

UX は ISO 9241-210:2010 で「a person's perceptions and responses that result from the use or anticipated use of a product, system or service」と定義されている<sup>[1]</sup>。すなわち、製品やシステム、サービスを利用した結果、または利用を想定した結果生じる人々の知覚および反応が UX である。これは、製品やシステム、サービスを「利用する前」、「利用している時」、そして「利用した結果」生じるユーザの気持ちや反応であると理解することができる。また ISO 9241-11/JIS Z 8521 では UX とは別に Quality in use (利用時品質) が「ある製品が、指定されたユーザによって、指定された利用の状況下で、指定された目的を達成するために用いられる際の、

---

<sup>1</sup> 事例提供：エスディーテック株式会社 鈴木 啓高 氏

有効さ、効率及びユーザの満足度の度合い」と定義されている<sup>[2][3]</sup>。製品の UX を向上するための一つの指針として製品の利用時品質を適用し、利用時品質の向上を実現することで UX の向上に貢献することができる。利用時品質を向上するために、我々は設計、開発、および開発後の各段階において様々な取組みを行う必要がある。

UX および利用時品質の向上に関しては従来より人間中心設計の手法が広く議論されていて、人間中心設計機構<sup>[4]</sup>の取組みなどで確認することができる。また近年ではデザインシンキングに関する取組みも数多くみることができる。これらの取組みに共通していることの一つとして、「ユーザ視点で物事を測る」ことの重要性がある。我々が UX および利用時品質の向上を目指すというのは、“ユーザが”製品を利用する際に生じる反応や、有効さ、効率及び満足度の度合いである。それゆえ、常にユーザ視点で検証、評価を行うことが重要であるというのは必然と言えるだろう（図 91-1）。

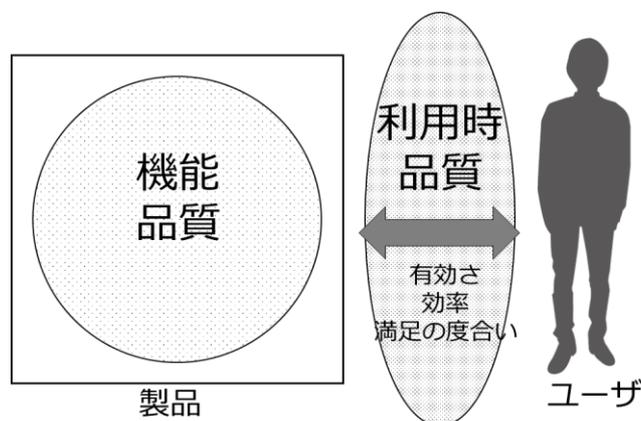
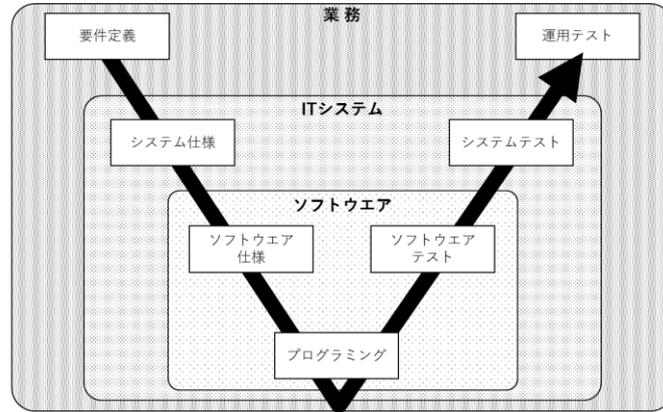


図 91-1 2つの品質

ユーザ視点での評価手法としてユーザビリティテストが実施されることが多いが、これだけでは UX の評価手法としては不十分である。ユーザビリティテストはユーザビリティラボや会議室という実験的環境で行われるものであるのに対して、UX はユーザが実際の製品やシステムやサービスを、実際の業務環境や生活環境のなかで利用して経験することである<sup>[5]</sup>。したがって、ユーザビリティテストそれ自体は非常に重要でかつ効果的だが、UX の評価は「実際の環境」で実施する必要がある。また、なんらかの評価を行った結果問題点が見つかった場合は速やかに改善を反映し、かつ評価、改善のプロセスを繰り返し実施する必要がある、これを可能とする仕組みも欠かせない。利用時品質の高い HMI を実現するためには、このような Agile 開発が求められている。

ところが、現在の自動車の HMI 開発の現場におけるプロセスはこういった取組みを実現することが非常に難しい状況にあると言わざるを得ない。多くの開発現場では細かな違いはあれど、図 91-2 に示したような、いわゆるウォーターフォール型、V 字型のプロセスで開発を進めている。



経営者が参画する要求品質の確保 ～超上流から攻めるIT化の勤どころ～独立行政法人 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 編 第3章 要求品質の確保に向けて 図3.1超上流プロセス

図 91-2 V字型開発プロセス

全体のプロセスを構成する各工程ではその工程のアウトプットを評価し、妥当性を検証した上で次の工程へと開発を進めていく。例えばデザインのアウトプットの検証としてユーザビリティテストを実施することは可能だが、機能実装に着手していない段階では「実験的環境」でのテストに限定されてしまう。開発の工程が進み、機能実装が完了した段階で比較的「実際の環境」に近い状態でようやく HMI の検証が行えたとしても、そこで見つかった問題を改善するためには相当の手戻りが必要となってしまう。これは多くの場合、開発スケジュールやコストの制約から現実的ではない。ましてや、評価、改善のプロセスを繰り返し実施することは不可能に近いという場合がほとんどであろう。

評価、改善のプロセスを繰り返すためには、デザイン工程と開発工程とをスムーズに往き来できることが欠かせない。ユーザビリティテスト等を経てデザインの工程が完了したら、そのアウトプットとしてデザイン指示書、デザイン仕様書といった資料を作成し、次工程である開発へとバトンタッチをする。デザイン指示書、デザイン仕様書は多くの場合、マイクロソフト社の PowerPoint、Excel などを利用して作成されている（図-91-3）。

The figure shows a design instruction document. On the left, there is a table titled 'XXXX デザイン規約 (ボタンサイズ)' (XXXX Design Guidelines (Button Size)). It specifies a minimum size of 18mm and shows two button designs: a rectangular one with 'wwwwwwww' and a square one with 'www'. Below the table are some notes in Japanese. On the right, there is a screenshot of a UI mockup for an iPod/iPhone interface. The mockup shows a sidebar with icons for 'Nav', 'Radio', 'Media', 'Youtube', and 'Climate', and a main area with a list of songs. A red box highlights a specific button in the mockup. Below the mockup is a table with columns for 'ID', '名称' (Name), 'UIコンポーネント' (UI Component), '状態' (State), '備考' (Remarks), '対応OS' (Supported OS), and '使用場所' (Usage Location). The table lists various UI components like 'Common\_Menu', 'AS', 'Header\_Button', 'iButton\_AID', 'Lボタン', and 'Common\_BS'.

図 91-3 デザイン指示書の例

結果として、HMI 開発に携わるデザイナーとエンジニアは、実際の HMI デザインそのものではなく、主にデザイナーが作成した Excel ファイルを通してコミュニケーションを図ることになる。デザイナーにとっては Excel ファイルを作るという追加工程が発生し、またデザイナーとエンジニアの両者にとって意図が正しく伝わらないリスクが発生する。このようにデザイン工程と開発工程とが直接的につながっていない環境においては、これらの工程間をスムーズに往き来することは難しいのが現実である。

我々は、ここにあげたような状況を改善し、HMI の評価、改善のプロセスを繰り返し実施できるようにするためには、以下に挙げる 2 点に取り組むことが重要であると考えている。これらはいずれも人間中心設計やデザインシンキングでもその重要性が示されている。我々はこれらの取り組みをプロトタイプ開発だけでなく、量産に向けた製品開発にもしっかりと適用していくという想いも込めて「デザインエンジニアリング」と呼ぶプロセスの中で定義している。

- (1) 機能実装に着手していない段階から、「実際の環境」に近い環境で HMI を検証できる仕組み作り
- (2) 機能実装に着手した後に発見した問題点の改善を、手戻りの影響を少なく実現し、繰り返し実施するための仕組み作り

また、これらの取り組みは既存の開発プロセスに対して適用する際に、出来る限り現場の負担を少なくすることが重要である。つまり、導入および運用にかかるコスト（費用的なコストだけでなく、学習コスト等も含めたコスト全般）を最小限に抑えることが欠かせない。

特に導入に際しての学習コストは重要となる。何らかの新しいツールを導入することで問題解決を図ろうとしても、人間が本来持っている「現状からの変更」に対する抵抗に対峙することとなり、なかなか期待通りの効果を得ることができない場合が多い。新しいツールを開発、導入する場合には、この点も十分に考慮に入れる必要がある。

### 3. 取り組みの対象

我々は前章であげた課題を解決するための取り組みの一環として、デザイン工程から開発工程において活用できるツール (TRITO® Linkage) を開発した。TRITO® Linkage は中間ファイルを介してデザイナーとエンジニアがそれぞれの作業を効率的に進行し、かつ両者間のコミュニケーションを容易にする各種ツール群で、(1) 中間ファイルを生成するための Adobe の各種ツール用プラグイン、(2) 中間ファイルを編集、ソースコードを自動生成するためのツール、そして (3) ランタイム用ライブラリから構成される。TRITO® Linkage では前章で挙げた課題を以下のようなアプローチで解決することを目指している。

- 実際の環境に近い環境で HMI を検証する

前述の通り、UX を評価するためには「実際の環境」で検証を行うことが求められる。ユーザビリティテストであれば、例えば紙芝居的な HMI を用意することでテストを実施することが可能であるが、UX の場合は実際に動作することが重要なポイントとなる。ここで「実際に動作する」というのは、必ずしも実際の車載器の環境でなければならないというわけではない。例えば、PC やタブレットで動作する HMI を用意し車両に持ち込み、評価を行うことも可能であり、実際にこのような評価手法が取られることは多いだろう。

ところが、厳密な意味で「実際の環境」を考えると、PC やタブレットを用いての評価では十分とはいえないことが多い。特に多くの描画処理を伴うような HMI の場合、HMI の実行結果が実行環境の処理能力に多分に影響を受けることもあわせて評価対象とする必要がある。

そこで TRITO® Linkage では、HMI のデザイン工程の段階で、速やかに車載器上で直接実行可能な HMI アプリケーションを生成する手段を提供することで、この課題の解決を目指す。具体的には、デザインデータから自動生成した中間ファイルを、専用のツールでソースコードに変換する。このソースコードをビルドすることで実際の環境で動作するアプリケーションを速やかに生成することができる。

- デザイン工程と開発工程をより密につなぎ、Agile 開発を実現する

Agile 開発をスムーズに進めるためには、各工程間のコミュニケーションをスムーズなものにする必要がある。具体的には、現在多く行われている Excel 等による資料を介したコミュニケーションを排除し、より直接的にデザイナーとエンジニアがコミュニケーションを取れる手段を提供する必要がある。ここで問題になるのは Excel 等のファイル形式ではなく、一旦作成したデザインを別のフォーマット（この場合 Excel）で再構築し、それを解釈し、また別のフォーマットで（ソースコード）再構築する必要があるというプロセスの問題である（図 91-4）。

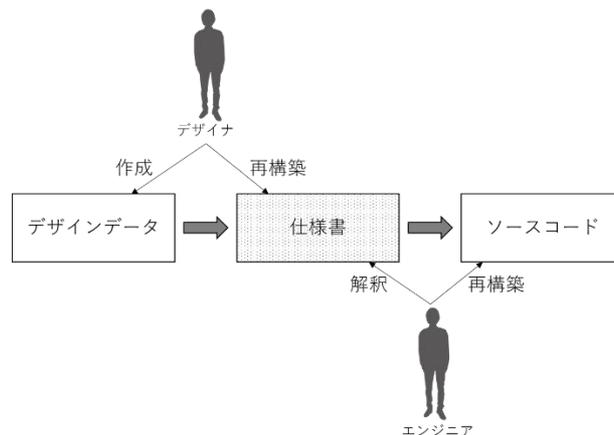


図 91-4 デザイン～実装の流れ

そこで TRITO® Linkage では、デザイナーとエンジニアの両方が直接参照可能で、かつデザインとソースコードの双方向に自動変換が可能な中間フォーマットを用意し、より直接的なコミュニケーションの実現を目指す。これにより、不要な工数の削減、デザインの意図の誤解の排除といった効果も目指す。

- 導入障壁を減らす

色々な課題はあるものの、開発の現場では慣れ親しんだ従来の手法が存在している。これを大きく変えて新しい手法を導入することは、その新しい手法の良し悪しに関わらず少なくとも抵抗があるものである。特にデザイナーが利用するツールは、デザイナーにとってのまさに手足の役割を果たすものであり、これを強制的に全く別のツールに変えることは現実的ではない。仮に全く別のツールを導入した場合でも、結局慣れ親しんだツールと併用する形になり、結果的に以前よりも工程が増えてしまったというケースもある。

そこで TRITO® Linkage では、デザイナーが慣れ親しんだツールについては出来るだけ従来通りのやり方で使い続けられる仕組みの実現を目指す。具体的には TRITO® Linkage ではデザイナー向けのツールを、Adobe の各種ツール(執筆時点では Photoshop、Illustrator、Animate、After Effects) 用のプラグインとして提供している。提供するプラグインは上記 Adobe ツールから直接利用できるものであり、したがってデザイン作業の流れを不必要に妨げることがない。

上記のアプローチで開発した TRITO® Linkage の効果を検証するために、実際の車載器向け HMI の開発資産を活用したプロジェクトを実施した。この取組みでは、既存のデザインデータおよび仕様書を起点とし、実機で動作するアプリケーションの実装までを TRITO® Linkage を活用して進めた。

作成する HMI は (1) 複数の出力ディスプレイ (センターコンソール、後部座席用スクリーン等) および、(2) 複数の入力デバイス (前方座席用コントローラおよび後方座席用コントローラ) を持ち、(3) メニューから選択可能なナビゲーション、ラジオ、オーディオ、エアコンをそれぞれ操作可能な画面を持つ (図 91-5)。

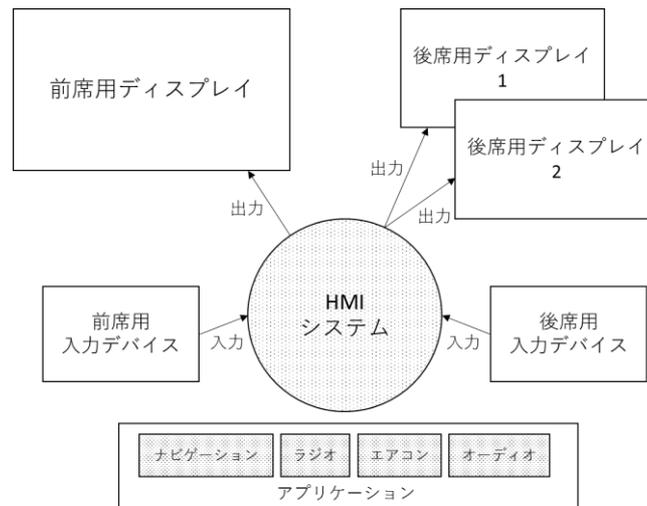


図 91-5 HMI の構成

基本的な画面レイアウトおよび画面遷移はすでに決まっているが、詳細な演出手法は検討中である。また、プロジェクト開始の段階では以下の各種データが用意されている。

- 画面レイアウトを記載した資料
- 画面遷移に関する情報を記載した資料
- 各画面毎に作成されたデザインデータ (Photoshop ファイル)

これらの情報を元にデザインを作り込み、アプリケーションを実装する必要がある。作成する HMI は実機で動作するアプリケーションとして実現する必要があるが、ナビゲーション等の各種機能は完全に実装が完了している必要はないが、紙芝居的な HMI アプリケーションでは不十分で、少なくとも機能実装を行う必要がある箇所が明確になっている必要があるものとする。一方で、車載器用の HMI で必要となる割り込み処理や状態遷移の管理（走行中には操作できないメニューの振舞いの管理等）についての対応は不要とする。

ツールの導入により「利用時品質の向上に向けた開発が実現できたか」どうかはどのようにして測ることができるだろうか。例えば、最終的に完成したアプリケーションの利用時品質を測り、その結果を比較することでツールの導入の効果を判定するという方法が考えられる。しかし、ツールはあくまでも利用時品質の向上をサポートするものであり、ツールの導入がただちに利用時品質の向上を実現するものではない。利用時品質の向上を実現するのはあくまでも適切な設計、デザイン、実装であり、これらを繰り返し評価、修正していくことによるのみ高い利用時品質が実現できる。

ツールに期待される役割は、これらの作業およびプロセスを効率的に進められるようにすることであり、その効果はツール使用者がどう感じたかという定性的な側面と、開発期間の短縮により評価、修正がより多く実施できたかという定量的な側面で測ることができる。ツール使用者

がツールに対して好意的な印象を感じれば、アウトプットへの好影響が期待できるだろうし、開発期間が短縮できれば同一の利用時品質をより短期間で実現できる。または同一の開発期間でより高い利用時品質を実現することができるといえるだろう。

そこで今回の取組みでは（１）ツール利用者の主観による定性的な評価と、（２）ツールを使わなかった場合の見積もり工数とツールを使って実際にかかった工数との比較による定量的な評価の２つの側面から、達成度を評価する。

#### 4. 取組みの実施、及び実施上の問題、対策、工夫

今回の取組みでは前述の通り、HMI デザインに関連する各種データがあらかじめ用意されている状態からのスタートとなる。本来であればコンセプト検討の段階から、その後の意匠作成、実装を見据えて連続性のある取組みとするべきである。具体的には、（１）デザイナー、エンジニアの意見もできる限り取り入れ、後工程で発生しうる問題を早い段階で明らかにし解決する。（２）上流工程の段階から積極的にツールを導入するなどしてデータの可搬性を高いものとすることが重要である（図 91-6）。特に（１）についてはデザインシンキングでも強調されている多様性の実現そのものであり、よりよい HMI デザインを実現し利用時品質を向上するためにも重要である。

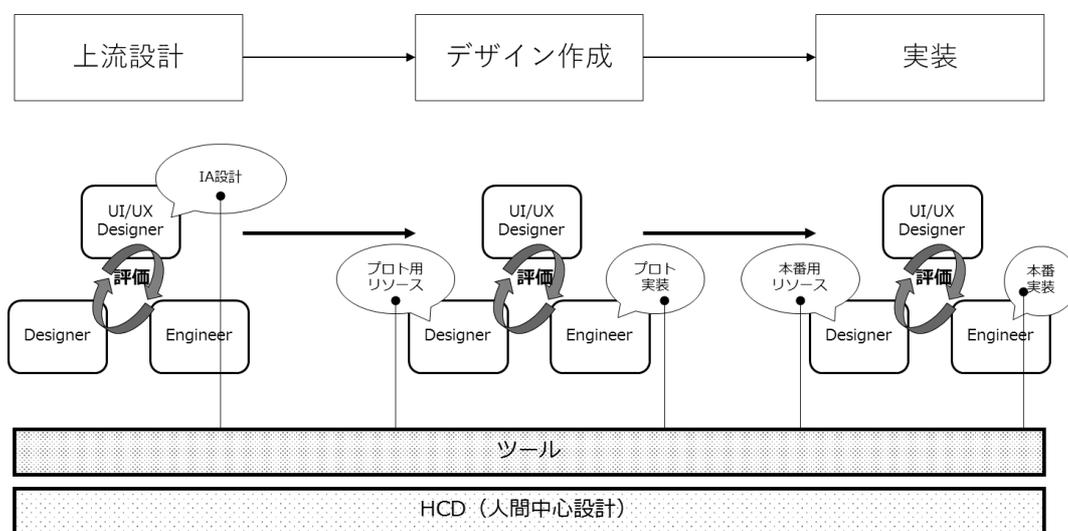


図 91-6 デザインエンジニアリングによる開発の流れ

しかし、実際の開発現場では当然既存のワークフローが存在し、ある時点でその全てを一気に変更するのは現実的ではない場合が多いだろう。今回の取組みでは基本となるデザインが既存のワークフローで作成されたところからの取組みとなるが、このように限られた範囲にツールの導入とワークフローの変更を適用することでも大きな効果が得られることが示せれば十分に意義があるといえるだろう。

今回の取組みの大きな作業計画は以下の通りである（図 91-7）

				1	2	3	4	5	6	7	8	9	10
	項目	内容	担当										
1	レビュー	用意されている資料の確認	全員	●	→								
2	変換	画面毎のデザインデータを修正	デザイナー			●	→						
		HMIアプリケーションの基本構成を作成											
3	機能実装	ソースコードの編集	エンジニア			●	→						
4	デザイン	デザインデータの編集	デザイナー				●	→					
5	結合	結合および評価	全員									●	→

図 91-7 開発スケジュール

作業計画：

- 1) 用意されている資料の確認：画面レイアウト資料、画面遷移資料、画面毎のデザインデータをデザイナーとエンジニアが確認し、全体像を共有する。今回の例では1つのHMIアプリケーションを複数の画面および複数のコントローラ（入力デバイス）が構成している。操作とそれに伴うHMIアプリケーションの挙動について、現時点で不明な点について設計担当に確認をする。
- 2) 画面毎のデザインデータを修正：用意されたデザインデータ（今回の場合はPhotoshopデータ）を確認し、今回適用するツール（TRITO® Linkage）で利用可能な形に（必要であれば）修正する。具体的にはPhotoshopデータのレイヤー構成およびレイヤー名をTRITO® Linkageがサポートする形に修正する。これはデザイナーがPhotoshop内で完結できる作業である。
- 3) HMIアプリケーションの基本構成を作成：Photoshopデータを修正したら、TRITO® Linkageの機能により、中間ファイルを自動生成する。ここで生成される中間ファイルにはレイアウトおよび各意匠だけでなく、より上位の概念であるUIコンポーネント（ボタン、リスト等）に関する情報および画面遷移に関する情報が含まれている。一旦中間ファイルを生成したら、TRITO® Linkageの機能により、生成した中間ファイルからソースコードを自動生成する。今回の取組みでは最終的なアプリケーション実行環境の構成にあわせてQt（Qt company社が提供するアプリケーション開発フレームワーク）<sup>6</sup>対応のソースコードを生成する。
- 4) デザインおよびソースコードの編集：デザイナーは意匠、アニメーションの作り込みに取組み、エンジニアは機能実装に取組む。これらの作業は独立して並行に進行することができる。デザイナーが作業の過程で実際の動きを確認したい場合は、その時点での中間ファイルを生成し、エンジニアが作業中の最新のソースコードに適用する。同様にエンジニアが作業の過程で最新のデザインを反映した動きを確認する場合は、その時点での最新の中間ファイルを取得し編集のソースコードに適用することで実現できる。つまりデザイナーとエンジニアが中間ファイルを介して常に最新の状態を必要に応じて確認することができる。このように“作りながら確認し、必要に応じて適宜修正を加える”ことにより、早い段階でHMIの問題を発見し解決することが可能となる。その際、デザイナーだけ、エンジニア

だけ、あるいはデザイナーとエンジニアだけで評価するのではなく、UI 設計者等の他のプロジェクトメンバーも評価に加わることが欠かせない。評価の結果、仕様の見直しが必要であるとなった場合には、直接の作業者の都合だけでなく、プロジェクト全体の視点で判断をし、速やかにその情報をプロジェクトメンバー全体で共有することができる。

- 5) 結合および評価：当初計画していた要件について実装が完了した段階を節目とし、正式に結合および評価を実施する。前述の通り、開発の過程で随時仕様変更が発生することがある。したがって、この節目についてもその段階で実装完了が求められる要件については随時変更が発生する。多くの場合日程の都合で節目を設定し、日程を制約条件として仕様変更の方法、可否を判断する必要があるだろう。

特に上記4)において、デザイナーとエンジニアとが並行して作業を進めるためにはツールの活用が欠かせない。また、並行作業の中で随時確認を行う際に、机上での確認にとどまらず、常に「実際の環境」にできる限り近い環境での確認を実現するうえでもツールが大きく貢献している。

今回の取組みでは上記の計画を10日間という非常に短い期間の中で実施した。実際の取組みは以下のように進行した。

#### 1日目～2日目：「用意されている資料の確認」

まずはデザイナーが受領した各種資料（画面レイアウトおよび画面遷移資料、各画面毎のデザインデータ）を確認し、このHMIアプリケーションでできること（機能要件）を再度整理した。整理した結果をエンジニアと共有、レビューしプロジェクトが開始した。この段階でデザインおよび実装上の問題点が発見された場合には速やかに仕様に修正を加える。今回は細かな課題はいくつか出たものの、仕様変更を要する規模の問題点は発見されなかった。本来であれば、さらに上流工程（アプリケーション企画検討、仕様検討）の段階でデザイナーおよびエンジニアを交えたレビューを行い、出来るだけ早い段階で問題を発見するようにする必要がある。今回は仕様検討が完了した段階でのレビューからの開始となったが、この段階でのレビューにデザイナーおよびエンジニアが積極的に関与することが重要である。これにより、早い段階での問題発見を可能とするだけでなく、デザイナー、エンジニアを含めた全てのプロジェクトメンバーに各自の担当範囲だけではなく、プロジェクト全体を“自分事”と感じさせるよう導くことが可能となる。

レビューが完了した後に、エンジニアは例えば「ナビゲーションの起動」、「音楽の再生」などの、HMIデザインに依存しない個別の機能モジュールの実装に先行して着手を開始した。

#### 3日目：「画面毎のデザインデータを修正」、「HMIアプリケーションの基本構成を作成」

続いて、HMIアプリケーションの実装を進めるための土台となるアプリケーションソースコードを用意する。今回の取組みではツールを活用しデザイナーがデザインデータから出力する中

間ファイルから自動生成した。エンジニアは自動生成されたソースコードを編集することで HMI アプリケーションの実装を進める。

ツールによりデザインデータから中間ファイルを出力しソースコードを自動生成するためには、元となるデザインデータがツールの想定するデータ形式になっている必要がある。具体的には Photoshop データのレイヤ構造、レイヤ名の命名規則が TRITO® Linkage の規定に従っている必要がある。そこでデザイナーは、まず受領した Photoshop データを確認し、必要に応じてレイヤ構造、およびレイヤ名を編集した。この作業は Photoshop データを Photoshop で直接編集するか、あるいは一旦中間ファイルを出力してから TRITO® Linkage が提供するツールで出力した中間ファイルを編集することで実現できる。今回の取組みでは受領した Photoshop データのレイヤ構造が元々 TRITO® Linkage の想定にほぼ従っていたため編集がそれほど必要とならないことが確認できたので、Photoshop データを Photoshop で直接編集した。

TRITO® Linkage ではこれまでの HMI アプリケーション開発の経験に基づき、レイヤ構造およびレイヤ名の命名規則を規定した。今回完全に外部のチームで作成された Photoshop データが大凡 TRITO® Linkage の想定するレイヤ構造にしたがっていたことは、この想定が十分に現実的で汎用的なものであることを示したといえるだろう。

必要な編集を行った後に中間ファイルを出力し、ツールにより Qt に対応したソースコードおよび Qt プロジェクトファイルを自動生成した。このソースコードには Photoshop データに含まれるレイアウトおよび意匠情報に加えて、より上位の概念である UI コンポーネントに関する情報も含まれている。例えばボタンを想定してデザインした矩形は、矩形という意匠としてではなく、ボタンオブジェクトとして実装される。したがって、自動生成したソースコードにはボタン等の UI コンポーネントに対するイベントハンドリング処理（ボタン押下時に実行される関数）も実装されている。

デザイナーがレイアウトおよび意匠データを作成する際には、当然 HMI デザインの構成要素として各種 UI コンポーネントを想定している。従来の Excel ファイルによるやり取りにおいては、レイアウトおよび意匠データを作成した後に、それぞれの意匠データがどのような UI コンポーネントを想定したものなのかを改めて整理し記述する必要があり、デザイナーにとっては二度手間の作業となっていた。今回ツールを活用することでこの手間を省き、デザイナーがより一層デザインに時間を割くことが可能となり、また、エンジニアは Excel ファイルからデザイナーの意図を解釈する必要がなくなった。

#### 4 日目～8 日目：「デザインおよびソースコードの編集」

中間ファイルと、そこから自動生成されるソースコードを共有した後はデザイナーとエンジニアがそれぞれ並行して作業を進行した。デザイナーはレイアウト、意匠、アニメーションを作り込み、エンジニアは機能モジュールの実装と HMI の実装を進める。

デザイナーとエンジニアはツールにより出力する中間ファイル、およびソースコードで最新の状態を共有しコミュニケーションを取る。

デザイナーは任意のタイミングで編集中のデザインデータから中間ファイルを出力する。出力した中間ファイルからツールによりソースコードを自動生成することで、デザイナー自身の手で実際の環境で HMI アプリケーションがどのように見えるかを確認することができる。中間ファイルからソースコードを自動生成する際には、既存のソースコード（過去に生成した中間ファイルから生成したソースコードをもとにエンジニアが編集を進めたもの）に自動的にマージすることができる。（図 91-8）したがって、デザイナーとエンジニアは常にそれぞれの最新の作業進捗を取り込むことができ、都度最新の状態のアプリケーションを動作させるためだけにお互いの手を煩わす必要がない。

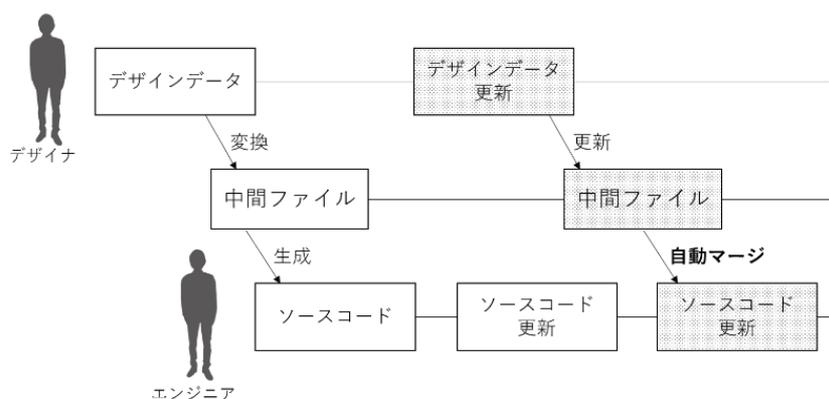


図 91-8 デザインとソースコードの更新

デザイナーとエンジニアのコミュニケーションを不要とするのではなく、コミュニケーションの実現にかかるコストを削減することで、より密なコミュニケーションを促すことができた。これにより常に「実際の環境」に近い環境で確認、評価、修正のプロセスを繰り返すことが可能となった。

8 日目にデザインおよび実装が完了した段階で一旦完了とし、最新の状態の中間ファイルおよびソースコードのマージを行った。

#### 9 日目～10 日目：「結合および評価」

プロジェクトメンバー全体で最終的な HMI アプリケーションの動作を確認して、細かな使い勝手を修正した。

規模が小さいとはいえ、10 日間という非常に短い期間での取組みを無事に完了することができた。次章では本取組みの振り返りと達成度の評価について述べる。

## 5. 達成度の評価、取組みの結果

当初期待した通り、デザインとエンジニアリングのプロセスを並行して進めることで短期間のプロジェクトを無事に完了することができた。利用時品質の高い製品開発を実現するためには「機能実装に着手していない段階から「実際の環境」に近い環境で HMI を検証できる仕組み作り」、および「機能実装に着手した後に発見した問題点の改善を、手戻りの影響を少なく実現し、繰り返し実施するための仕組み作り」が欠かせない。そこで本取組みではツールの導入によってこれらの仕組みの実現を試みた。結果として「実際の環境に近い環境で HMI を検証」を繰り返し、「デザイン工程と開発工程をより密につなぎ、Agile 開発を実現」することができた。また、基本的にはデザイナーおよびエンジニアともに既存の使い慣れたツール（今回の場合はデザイナーが使う Photoshop およびエンジニアが使う Qt 開発環境）をそのまま使うことが可能であったので、ツールの導入に伴い発生する学習コストも低く抑えることができた。

実際にプロジェクトに携わったデザイナー、エンジニアからフィードバックは以下の通りであった。

- (1) 新しいツールを導入することの負担はほとんどなかった（デザイナー）
- (2) アニメーションの詳細を文書化することなく、そのままアニメーションデータとしてエンジニアに渡せばよいのは楽だった（デザイナー）
- (3) 作成したデザインをすぐにアプリケーションとして動作させ確認することができてよかった（デザイナー）
- (4) ツールの動作が重かったり、思ったように使えなくて困ったことがあった（エンジニア）
- (5) デザインについてはほとんど気にせず、機能実装に集中することができた（エンジニア）
- (6) コードが自動生成、自動編集される際の変化点が明確になっていないと規模が大きくなった場合に困ると感じた（エンジニア）
- (7) 実際に動くアプリケーションを見ながら修正を繰り返すことで、最終的なアプリケーションの具体的なイメージを持つことができてよかった（デザイナー、エンジニア）

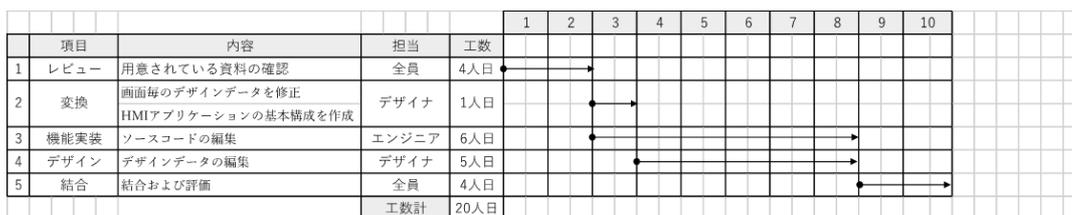
概ね良好なフィードバックが得られたと言えるだろう。特に「実際の環境に近い環境で HMI を検証」し、「デザイン工程と開発工程をより密につなぎ、Agile 開発を実現」する開発プロセスに関しては、開発効率の面だけでなく、上記（7）のように意識面での効果もあった。より具体的なアプリケーションイメージをプロジェクトメンバーが共有することで、デザインおよびエンジニアリングそれぞれに閉じた視点ではなく、よりユーザ視点に近いアプリケーション全体を意識した開発を全ての工程で実施することが可能となり、結果としてより高い利用時品質を実現することが可能となる。

ツールの導入に関しても良好なフィードバックが多かった一方で、ツールそのものの動作に対する不満もあった。開発中のツールのため一部機能に制限があったのは事実だが、やはり従来のやり方に対して強制的にツールを導入し、作業フローを変更することに対する抵抗は少なく

ないと言える。出来るだけ従来の作業フローからの移行をスムーズなものになるようなプロセスを構築し、また、ツール自体の利用時品質（この場合の利用者はデザイナー、エンジニアとなる）を向上することが欠かせない。

また、ツールで自動化される部分とそうでない部分とについて、明確に分離をしておく必要性についてのコメントもあった。特に大規模プロジェクトの管理を行う上では非常に重要な点であるだろう。プロトタイプ開発ではややもすると「動きさえすればよい」と判断されてしまう場合もあるが、量産向けのアプリケーション開発ではこれは許されない。そのためにも利用時品質だけでなく、製品品質・機能品質についても十分に検証、保証できる仕組みが必要である。特に Agile 開発で進行する場合にはテストの自動化などの仕組み化が欠かせないだろう。

今回はツールを導入することでデザイン工程と開発工程を並行して進行し、全体を Agile 開発で進めた。結果として 10 日間という短期間で完了することができたが、従来のウォーターフォール型で進行した場合の開発工数の見積りと比較してどうだっただろうか。（図 91-9）



今回手法の見積もり



従来手法の見積もり

図 91-9 見積もり比較

開発期間はウォーターフォール型に対して約 23%の削減、また開発工数は変わらずという結果となった。コスト削減の効果はあまり得られなかったといえるだろう。しかし、従来手法では「実際の環境に近い環境での HMI 検証」が 1 度しか行えないのに対して、今回のツールを導入した手法では任意のタイミングで随時行うことができています。また、従来手法では 1 度しか行えない検証の結果、何か改善を要する事項が見つかったとしても対応する時間が残されていない。利用時品質を向上する上ではこの点が非常に重要で、同じ期間の中でより多く検証、修正を繰り返せる今回の取組みに優位性があると考えている。

## 6. 今後の取組みの考察

人間中心設計、デザインシンキング、デザインエンジニアリングなどの考え方、取組みの重要性については開発現場の多くの人が賛同をしてくれる。また、利用時品質を向上することの重要性についても同様に賛同し、「今後はこのような取組みを進めていく必要がある」というフィードバックを得ることが多い。ところが、実際には開発現場ではまだまだこのような取組みはなかなか現実のものとはなっていない。特に量産開発になればなるほどにその傾向は顕著である。大規模な開発プロセスを変えていくには時間がかかるという理由もあるだろうが、このような取組みを量産の開発現場に適用していくにはまだまだ解決しなければならない課題があると言える。Agile プロセスの導入については他で多く議論されているので、ここでは特にツールの導入に関して、量産開発のような大規模開発に適応する際に顕在化する課題について以下に何点か挙げる。

- 大規模な HMI 開発へのツールの適応力

プロトタイプと違い、量産開発では数百画面、場合によっては 1000 画面の規模で構成される HMI を開発する必要がある。画面遷移、状態遷移も複雑になり、これを効率的に抜け漏れなく管理する必要がある。また、ここでいう状態遷移には HMI の状態遷移だけでなく車両の状態遷移も含まれる。例えば「走行中はこのメニューは操作できない」、「音楽再生中に危険通知の表示を表示する」といった制御をツールでデザイン・実装できるだけでなく、システム全体として抜け漏れがないことを管理できることが求められる。

- 大規模な開発チームへの適応力

HMI に限らず、量産開発のプロジェクトは全世界にデザイン・開発メンバーが分散していることが多い。特にデザインエンジニアリングにおいては、開発中のプロジェクトチーム内での密なコミュニケーションが欠かせない。必ずしも席を隣り合わせている、あるいは隣の建屋に行けば顔を合わせられるといった状況にない中で、デザイナーおよびエンジニアが進行中の成果物を共有し、密にコミュニケーションを取りながら開発を進行するための手段が必須となる。

- 製品品質・機能品質への適応力

プロトタイプ開発においてももちろん品質は重要であるが、量産開発で求められる品質は比較にならないと言えるだろう。特に品質が高いだけでなく、万が一何か問題があった場合のトレーサビリティを担保することが重要であり、これを実現することがツールにも求められる。特に車載向け HMI は、その適応領域が従来のナビゲーション、オーディオなどからスピードメータなどの計器類にまで範囲を広げている。そのような領域においては機能安全の国際標準規格である ISO26262<sup>[7][8][9][10][11][12][13][14][15][16][17][18]</sup>への対応も求められる場合があるだろう。

今後は特に上記に挙げた点をツールでいかに対応していくか検討および検証し、デザインエンジニアリングを量産開発へ適応する取組みを、より一層充実させていきたいと考えている。

参考文献

- [1] ISO9241-210:2010, <https://www.iso.org/standard/52075.html>
- [2] ISO9241-11, <https://www.iso.org/standard/63500.html>
- [3] JIS Z8521, <http://www.jisc.go.jp/jis-act/reading.html>
- [4] 人間中心設計機構, <https://www.hcdnet.org>
- [5] 黒須正明, 「UX への大いなる誤解」, [https://column.prime-strategy.co.jp/archives/column\\_1261](https://column.prime-strategy.co.jp/archives/column_1261)
- [6] Qt, <https://www.qt.io/what-is-qt/>
- [7] ISO/DIS26262-1, <https://www.iso.org/standard/68383.html>
- [8] ISO/DIS 26262-2, <https://www.iso.org/standard/68384.html>
- [9] ISO/DIS 26262-3, <https://www.iso.org/standard/68385.html>
- [10] ISO/DIS 26262-4, <https://www.iso.org/standard/68386.html>
- [11] ISO/DIS 26262-5, <https://www.iso.org/standard/68387.html>
- [12] ISO/DIS 26262-6, <https://www.iso.org/standard/68388.html>
- [13] ISO/DIS 26262-7, <https://www.iso.org/standard/68389.html>
- [14] ISO/DIS 26262-8, <https://www.iso.org/standard/68390.html>
- [15] ISO/DIS 26262-9, <https://www.iso.org/standard/68391.html>
- [16] ISO/DIS 26262-10.2, <https://www.iso.org/standard/68392.html>
- [17] ISO/DIS 26262-11, <https://www.iso.org/standard/69604.html>
- [18] ISO/DIS 26262-12, <https://www.iso.org/standard/69605.html>

掲載されている会社名・製品名などは、各社の登録商標または商標です。

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター (IPA/SEC)