

66

システム全体の再構築から始める派生開発の効率化¹

～自動車：電動パワーステアリング開発での適用事例～

1. 概要

要求の多様化・複雑化、ならびに市場投入の短納期化が進む昨今の製品開発において、派生開発の置かれている環境は厳しくなる一方である。当社の自動車部品事業もその例外ではなく、製品のバリエーションが増大することによる品質・コストへの影響が無視できないレベルになっている。そこで日本精工では、将来にわたってこの状況に対峙していくために、全製品のベースとなる「標準成果物」をソフトウェアのみならず、システムレベルとなる上流工程から整備し、それを用いて効率的な派生開発を行う「プロダクトライン開発」ができる集団になることを目指している。この標準成果物＝コア資産を新規開発するにあたり、派生開発向けに提唱されている USDМ (Universal Specification Describing Manner) や PFD (Process Flow Diagram) といった各種手法²を実践している。

本編では、これらの手法を実践する上で工夫した点や得られた気づき、今後の展望を紹介する。

2. 取り組みの目的

2.1 取り組み対象製品およびビジネス環境

本編で取り上げる製品は、自動車向け電動パワーステアリング (Electric Power Steering : 以降、EPS) である。車両の電動化や低燃費化に伴い、パワーステアリングも従来の油圧式から電動式への移行が進んでおり、昨今の乗用車における採用はほぼ電動式となっている。電動式は自動運転の実現にも必須の技術となっており、今後もその市場規模は拡大が続くと見込まれる。

当社はこの市場に独立系のシステムサプライヤとして参入しており、国内外の複数のカーメーカー、および複数の車種向けに製品を納入している。すなわち、「拡大の続く市場に対し、多数のバリエーションとなる製品を同時に開発している」というビジネス状況となっている。

2.2 解決すべき状況

このビジネスを今後も継続していく上で、昨今生じている開発コストの上昇傾向が強く懸念されている。

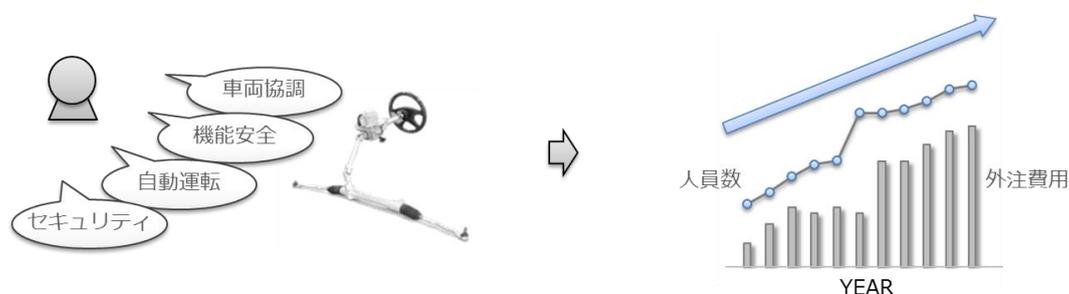
今や EPS は車両の「曲がる」を単体で実現するシステムではなく、車両全体としての複雑な

¹ 事例提供: 日本精工株式会社 須永 毅 氏、高橋 寛之 氏

² 本編ではこれら手法そのものについては説明しない。[1][2][3]を参照のこと。

協調動作を担うシステムであることが求められている。例えば自動駐車支援システムやレーンキープアシストなど、ドライバーによる直接のステアリング操作以外の制御系統と協調する機能が既に商品化されている。これらは自動運転技術へも繋がる技術となっており、また車車間・路車間といったネットワーク接続の高度化と共にセキュリティ対応の強化も求められ、今後ますます要求が多様化・複雑化していくことが見込まれる。さらに近年では自動車機能安全規格「ISO 26262」への対応も必須となっており、ただ機能を高品質に実現するだけでなく、その実現過程における組織の成熟も欠かせない要求となっている。このように、EPS 開発に求められる要求の多様化・複雑化が刻々と厳しくなっている状況である。

この背景を一因として、開発コストの増大が続いている。ある期間のソフトウェア開発に関わる人員数、およびその人員の工数不足を賄う外注費用の変遷を見ると、右肩上がりの増加が実際に起きていることが確認できた。このイメージを図 66-1 に示す。



EPS に求められる市場要求が厳しくなっている。これを背景に開発コストの増大が起きており、開発に携わる人員数、および外注費用が右肩上がりになっている傾向が確認できた。

図 66-1 要求の多様化・複雑化に伴うコスト増大イメージ

こうした要求の多様化・複雑化は、コスト増に対する外部要因としての位置づけである。その他に内部要因として、当社のソフトウェア開発スタイルも一端として存在すると考えられる。

複数の製品を同時に実現する上では、既存の製品をベースとした派生開発が行われているが、これがカーメーカーごと、さらには車種ごとの「一品一様」で進化しており、ビジネスの拡大に応じてそのバリエーションが増え続ける傾向になっている。この派生の様子はこれまで収束する気配を見せたことがなく、このままのスタイルを継続するとバリエーションが発散し、開発現場に破綻を招いてしまうことが懸念される。この様子を図 66-2 に示す。

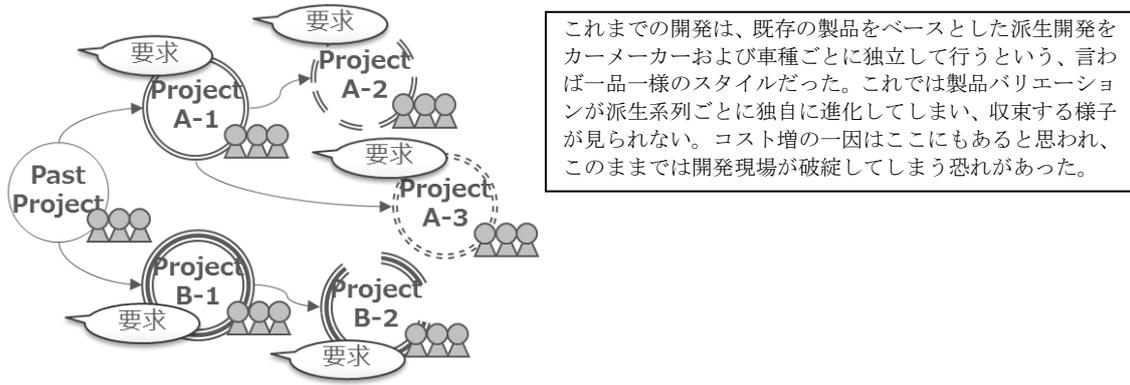


図 66-2 これまでの開発スタイル

2.3 目標

このような開発スタイルを取り続ける限り、状況が大きく改善されることはない。与えられる市場要求に「ありもの」を使って都度応えていくやり方ではなく、システムサプライヤとして自社製品の「標準成果物」を形成し、市場要求との差分を開発することで製品を実現していくスタイルに変えることが必要ではないかと考えた。すなわち「一品一様」の開発スタイルから脱却し、標準成果物を「コア資産」としたプロダクトライン型の開発を行える組織に変革して行くことが、本活動の目標である。この様子を図 66-3 に示す。

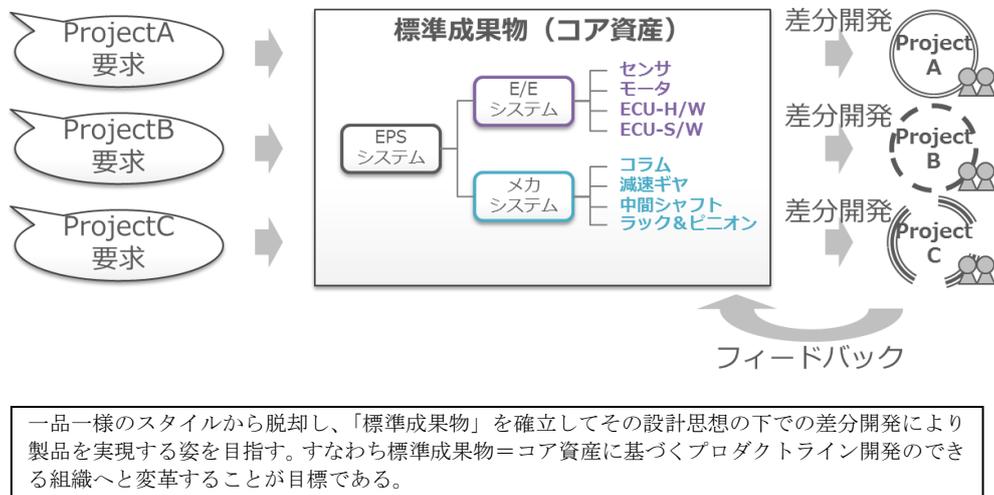


図 66-3 目指す開発の姿

なお標準成果物としての作成対象は、ソフトウェアのみではなく「EPS システム」全体とした。ソフトウェアはあくまで製品の一部に過ぎず、システムサプライヤとしてはシステム全体を開発しているからである。そこでシステム全体の構成要素を図 66-4 に示すように階層的に明確化し、標準成果物をそれぞれの階層で形成していくこととした。

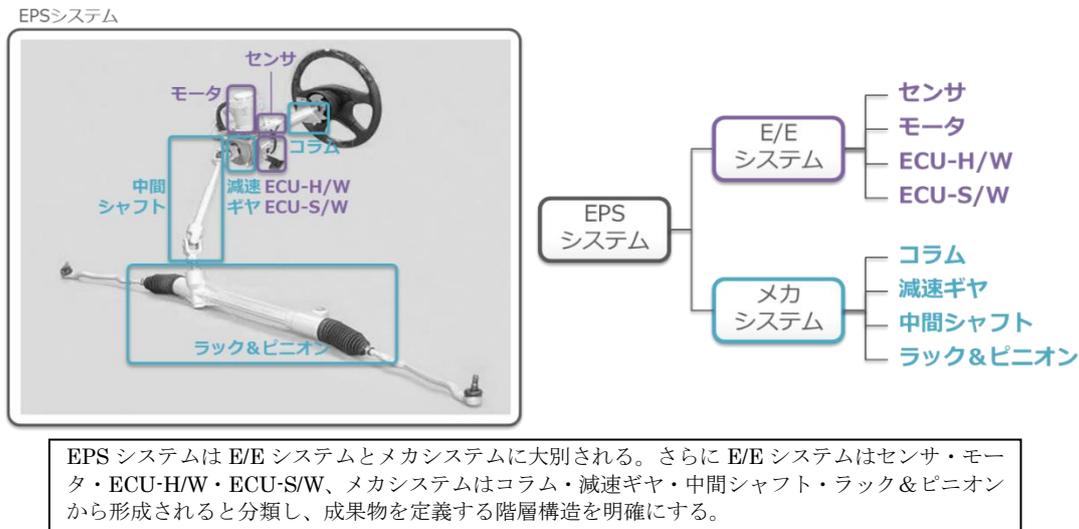


図 66-4 システム構成を階層構造で定義

3. 取り組み方針

3.1 取り組み前の課題

プロダクトライン型の開発を目指すとしたときに、現状の成果物および開発スタイルから解決すべき課題がいくつか抽出された。以下にそれらを説明する。

3.1.1. 課題①：要求の明確化

これまで製品を実現してきた既存の成果物内に、製品の各構成要素が実現すべき「製品仕様」は存在している。ソフトウェアを例に取れば「ソフトウェア機能仕様書」として、何を実装する必要があるかは明文化されている。また当然ながら、カーメーカーから受領する生の声としての「顧客要求」も存在する。

しかし、顧客要求をソフトウェア機能仕様書につなぐ記述、すなわちソフトウェアに対して何が求められるかという「ソフトウェア要求」の明文化は不足していた。よって既存の資料やソースコードを見渡しても、なぜその機能が必要なのか、なぜその実装が求められるのかという背景が不明確な箇所があった。標準成果物を構成しようとする上で、「なぜそれをするのか」がわからないままという状態は、大きな障害となり得る。

また、システムの上位階層になるにつれ、既存の成果物が乏しくなっていくという傾向も見られた。ソフトウェアやハードウェアであれば、技術者が直接設計・実装している対象であるため、先に述べたような仕様書はまだ存在している。しかし1段階抽象化された「E/Eシステム」のレベルになると、直接のモノづくり対象ではないこともあり、成果物としての品揃えの不足や内容の希薄さが散見された。システム全体の標準成果物を形成しようとする上では、上位階層の成果物もちろん必要となる。それぞれの階層において要求を明確に導き出すことで、標準的な要求およびそれに相応しい仕様の導出につなげていく必要がある。

3.1.2. 課題②：共通・固有の見極めと表現

複数のカーメーカー向けに製品を開発する上で、既存の成果物は製品ごとに独立して作成されている。これを見渡して整理する視点が存在しないため、何が製品間で共通的に実現している機能・技術なのか、何が特定のカーメーカー固有に用意しているものなのか、といった整理が存在していない。前者は再利用可能な共通性として、後者はバリエーション開発における可変性として捉えることで、有益な標準成果物を形成していくことができると考えられる。これを見極め、どう表現および管理をしていくかを検討する必要がある。

ここまでで挙げた「課題①：要求の明確化」と「課題②：共通・固有の見極めと表現」は、いずれも“要求仕様書構成”の問題であると捉えられる。この様子を図 66-5 に示す。

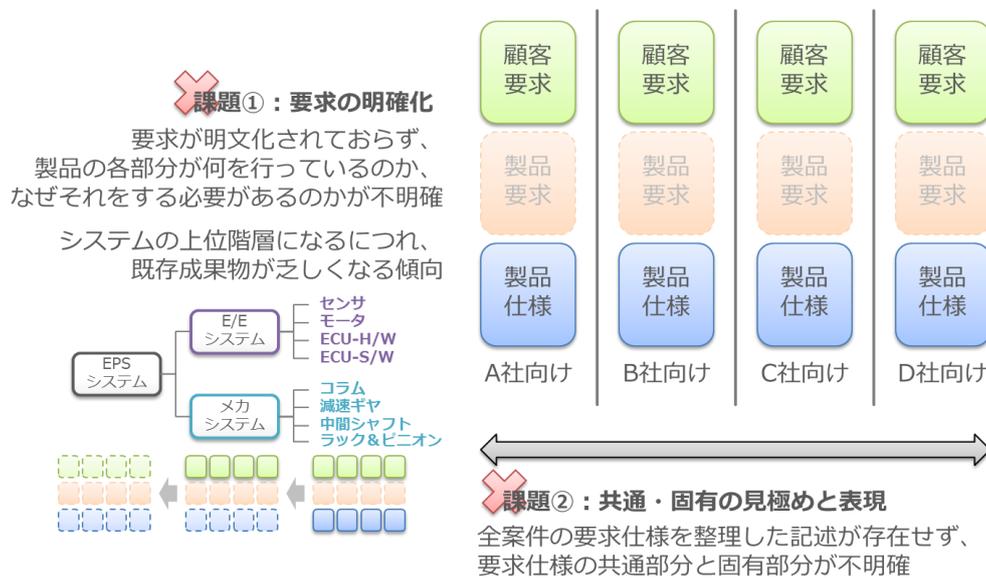


図 66-5 要求仕様書構成としての課題①および課題②

3.1.3. 課題③：上位階層の開発プロセス

課題①で述べたように、上位階層になるにつれ既存成果物が乏しくなっていく傾向がある。これはすなわち、上位階層の開発プロセスが元々明確化されていないため、成果物にばらつきが生じていることを意味する。標準成果物としては上位階層のものも揃えて行きたいが、その進め方について過去事例を参考にすることができないため、本活動の中で新たに開発プロセスを明確にしていく必要がある。

3.1.4. 課題④：標準成果物を用いた開発プロセス

目標とする開発スタイルでは、標準成果物を基にして製品を実現しつつ、そこで見られた新たな共通性を標準成果物へフィードバックして行く。このスタイルでの開発実績はこれまでに存在せず、本活動によって新たに確立するものとなる。その開発プロセスを明確にしていく必要が

ある。

ここまでで挙げた「課題③：上位階層の開発プロセス」と「課題④：標準成果物を用いた開発プロセス」は、いずれも“開発プロセス構築”の問題であると捉えられる。この様子を図 66-6 に示す。

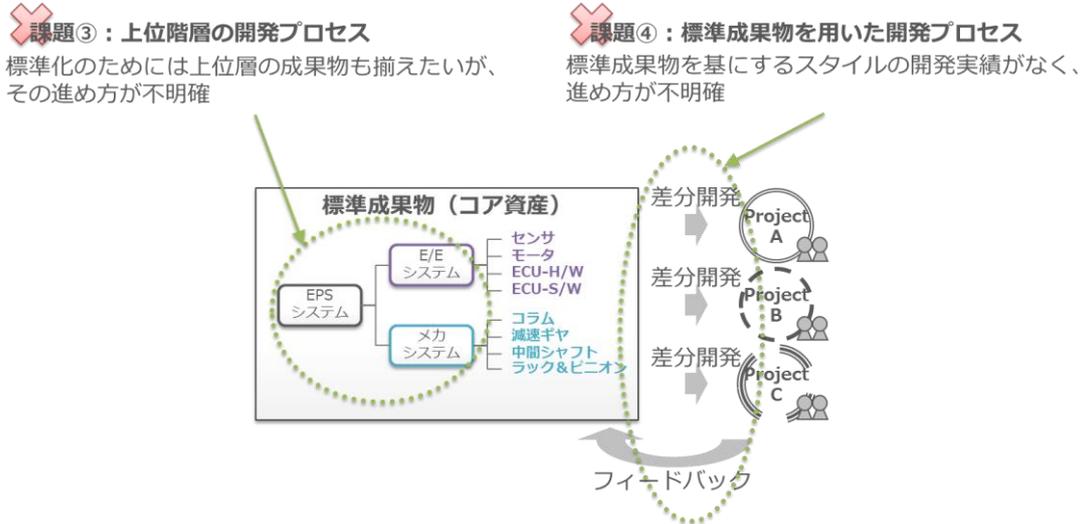


図 66-6 開発プロセス構築としての課題③および課題④

3.2 採用した方法論

「3」で述べた課題、すなわち“要求仕様書構成”としての「課題①：要求の明確化」と「課題②：共通・固有の見極めと表現」、「開発プロセス構築」としての「課題③：上位階層の開発プロセス」と「課題④：標準成果物を用いた開発プロセス」に対応できる方法論として、前者へは USDM、後者へは PFD をそれぞれ採用した。以下にその理由を記載する。

3.2.1. 要求仕様書構成：USDM

USDM はシステムクリエイツの清水吉男氏によって提唱された要求仕様書の記載マナーである[1][2]。この持つ特徴が、要求仕様書構成における課題①②の解決に適していると考えた。その様子を表 66-1 に示す。

表 66-1 USDM の特徴と採用理由

特徴	課題への採用理由
「要求」と「仕様」を明確に区別し、階層構造で整理する	課題①：要求の明確化 仕様だけでなくその背後の要求を明確にしていく必要があるため、これらの区別を大前提とする点が整理の上で有益 EPS システムの各階層の要求仕様書を明確にしていくため、階層構造を特徴として持つ点がまとめ方に有益
要求とともに「理由」や「説明」を記載する	課題①：要求の明確化 要求を明確にしていく上で、その背景にある「理由」を明記することは、要求内容の理解（なぜそれをやりたいのか）、共通性を見極め（やりたいことは同じなのではないか）、適切な仕様の導出（このやり方が本当にふさわしいのか）などに有益
Excel を基本とした表フォーマットで記載する	課題②：共通・固有の見極めと表現 Excel は広く普及しているツールであり、かつ行・列の追加の自由度が高い。USDM でもこの点をメリットとして挙げており、応用例として Traceability Matrix[1]の実現が見られる。共通・固有を見極めて表現する手段として有益

3.2.2. 開発プロセス構築：PFD

PFD も USDM と同じくシステムクリエイツの清水吉男氏によって提唱されているプロセスを設計するための技法である[3]。「プロセス」は1つ以上の「成果物」を入力・および出力する作業であると位置づけ、それらの連鎖をフロー状のダイアグラムで表現する。また「プロセス記述」と「成果物定義」を合わせて用意することで、それらの内容も明確に定義される。

課題③および課題④では、それぞれこれまで存在しなかった開発プロセスを明確にする必要があるため、本手法が有益であると考え、採用した。

4. 取り組み事例：USDM を用いた要求仕様書構成

4.1 要求の明確化

課題①：要求の明確化では、「仕様に対して要求が明確化されていない」と「上位階層ほど既存成果物が乏しい」の2つの問題点を内包していた。これらを USDM による整理で解決していく上で、2方向のアプローチを取る。

「仕様に対して要求が明確化されていない」ことに対しては、ボトムアップのアプローチを取る。すなわち、現存する成果物の仕様記述から、その仕様が必要とされる要求を逆方向に導き出すことで、これまで明らかにされていなかった暗黙知の部分を明確にしていく。なおこのアプローチは、すでに仕様の記載された成果物（ソフトウェア機能仕様書）が存在するソフトウェア開発に対して行う。

「上位階層ほど既存成果物が乏しい」ことに対しては、トップダウンのアプローチを取る。EPS システムに求められる要求を改めて議論・整理し、それを入力として EPS システム要求仕

様→E/E システム要求仕様→ソフトウェア要求仕様、といったように下りながら明確にしてい
く。

以下、それぞれのアプローチの中身を述べていく。

4.1.1. ボトムアップアプローチによる現状の整理

このアプローチでは、既存の「ソフトウェア機能仕様書」を入力とし、以下の手順に従って要求仕様を整理していく。

Step1. 既存の機能仕様書（仕様の記載はあるが、要求は不明確）に書かれている内容を、USDМの「仕様」として抽出する

Step2. 「仕様」から、それが求められる要求およびその理由を導き出し、明文化する
この様子を図 66-7 に示す。

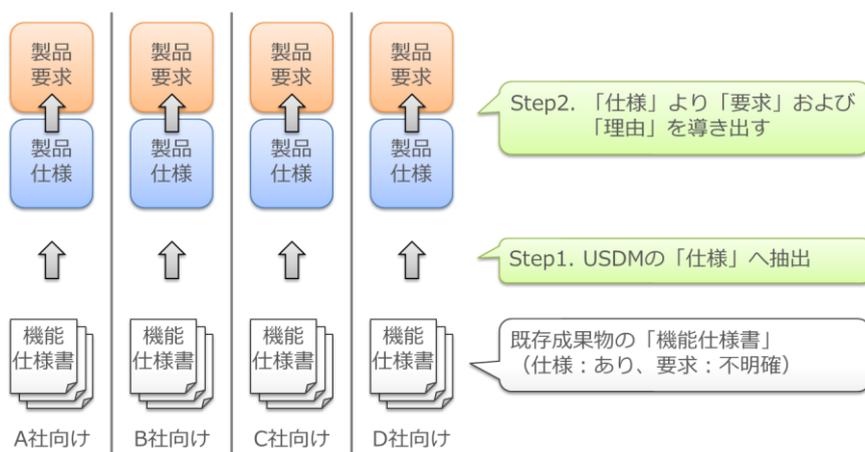


図 66-7 ボトムアップアプローチによる現状の整理

この手順で要求・理由・仕様の導出に成功すると、USDМのフォーマットとしては項目が埋められる形となり、やっていること（仕様）だけでなくその背景となる要求を理由とともに残すことができる。この様子を図 66-8 に示す。

制御車速算出	要求	VEHICLE-R-04.02.41	実際には起こりえない急激な車速の変化を緩和し、違和感の少ない車速情報を算出する。	製品要求 ↑ 製品仕様 ↑ 機能仕様書
	理由		急激な車速変化が発生すると、過剰なアシスト制御が行われるため。	
	説明			
	□□□	<変化割合制限> VEHICLE-S-04.02.401		

既存の機能仕様書にはやるべきことが書いてあるため、それを USDМの「仕様」部分へ抽出する。
何をやりたいのか（要求）、なぜやりたいのか（理由）は仕様から逆方向に類推して明文化する。
※仕様部分は製品具体情報のためぼかし処理

図 66-8 「要求」「理由」「仕様」が導出できた例

ただし常にそう上手くいくとは限らない。仕様から要求を導出してはみたものの、なぜそれをしたいのかという理由がわからないままとなる例も存在する。その場合は「理由」欄が埋められずに残る。しかしこの場合は何が不明確なのかを明らかにできたことになる。この様子を図 66-9 に示す。

車速上昇	要求	VEHICLE-R-03.12.23	車速が閾値を超えた場合、エンジンは作動していると判定する。	製品 要求
	理由			
	説明			
	□□□	<車速の上昇> VEHICLE-S-03.12.221		製品 仕様
				機能 仕様書

「仕様」から何をやりたいのか（要求）は考えられたものの、なぜそれをやりたいのか（理由）がはっきりしない場合の例。ただし何がわからないかを明確にすることはできており、ヒアリング等で有益となる。

図 66-9 「仕様」から「要求」を導出したが、「理由」が不明確なままの例

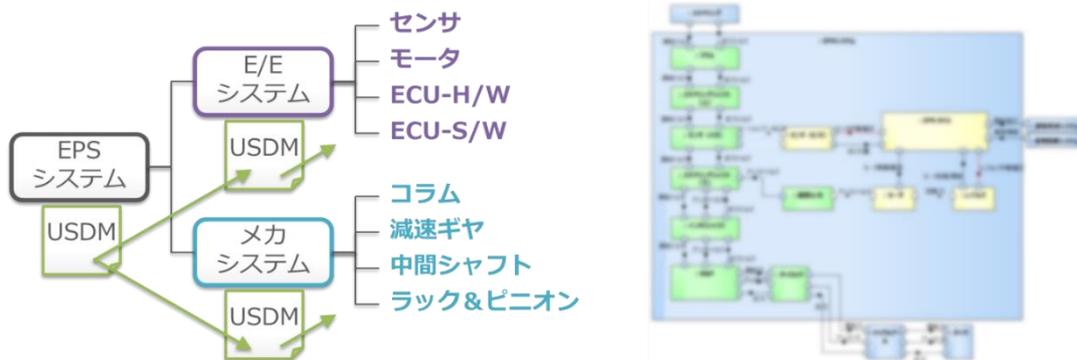
この取り組みにより得られた効果は以下である。

- 導出がうまく行った場合、これまで暗黙知とされていたことを形式知として表現することができた。
- 仮に導出できないものがあつた場合においても、不明確な点にポイントを絞って関係者にヒアリングするなど、無駄のない情報収集につなげることができるようになった。

4.1.2. トップダウンアプローチによる整理

このアプローチでは、構成要素の最上位である EPS システムという階層から、次の階層となる E/E システムおよびメカシステム、さらにその下の階層、というように要求仕様をまとめることを行う。

その際、上位階層の要求仕様書はこれまで不明確だったことから、その記載粒度について方針を決める必要がある。そこで今回は以下のような考え方で進めることとした。USDМ の階層数を第 1 要求→第 2 要求→仕様の 3 階層で構成するとするならば、最下層となる第 3 階層（仕様）の記述において、下階層で行うべき仕事の分配ができる記載粒度とする。そのために各階層では、該当階層の要求に基づくアーキテクチャを見出し、その構成要素を洗い出す。最下層の記述にその構成要素を割り当てられるよう、要求仕様の記載を分解して行く。この様子を図 66-10、USDМ の記載イメージを図 66-11 に示す。



各階層に与えられる要求を基にアーキテクチャ（右）を見出し、そこで出てくる構成要素へ仕事が分配できるよう、USDM 記載をまとめていく。

図 66-10 トップダウンアプローチにおけるまとめ方針

最下層記述をアーキテクチャ構成要素へ分配

EPSシステム		E	E	センサ(E/E)	EPS ECU
理由	ドライバのステアリング操作によって生じる操舵トルクを検出し、ドライバが意図したタイヤ角となるように、ナックルアームへの推力を発生させる。				
説明	ドライバが楽な操作で思い通りに車両を操舵できるようにするため。 「EPS教育資料」にて、一連の動作を説明するアニメーションがある。				
<ドライバ操舵の検出>					
要求	EPS.01.01	ステアリングから入力される操舵トルクの大きさと方向を検出する。			
理由		ドライバの操舵内容に応じて適切なアシストトルクを決めるため。			
説明		—			
<ステアリング操舵トルクの入力>					
<操舵トルクの検出>					
□□□	EPS.01.01.11				
□□□	EPS.01.01.12				
□□□	EPS.01.01.13			●	
□□□	EPS.01.01.14				●

E/Eシステム			
理由	EPS.01.01.13	トーションバー入力軸、出力軸の位置を測定する。	
説明		ねじれ量の推定をするには量軸の位置が必要となるため。	
<入出力軸位置の測定>			
要求	EPS.01.01.13.01	トーションバー入力軸、出力軸の位置を測定する。	
理由		ねじれ量の推定をするには量軸の位置が必要となるため。	
説明		—	
<位置検出手段>			
□□□	EPS.01.01.13.01.01		
□□□	EPS.01.01.13.01.02		

要求から見出したアーキテクチャの構成要素を USDM の Traceability Matrix（右端の列）として取り、最下層記述（仕様）と構成要素の対応を明確にする。最下層記述は下階層への要求となり、同様な手法で要求仕様が分解されていく。

図 66-11 トップダウンアプローチにおける USDM 記載例

この手法でまとめることで得られた効果は以下である。

- これまで不明確だった上位階層の要求仕様を明確化することができた。
- その記載粒度、およびアーキテクチャとの対応を明確にすることが可能となった。

4.2 共通・固有の見極めと表現

課題②：共通・固有の見極めと表現については、既存成果物を基にまとめていく「4」のボトムアップアプローチの結果をベースとして、以下の手順で要求仕様を整理することで解決を狙う。

Step1. ボトムアップアプローチにより製品ごとの要求仕様が導出されるため、それを横並びにして俯瞰する

Step2. 俯瞰した内容から製品に共通する要求仕様、製品固有の要求仕様の区別を見出し、それを表現する

この様子を図 66-12 に示す。

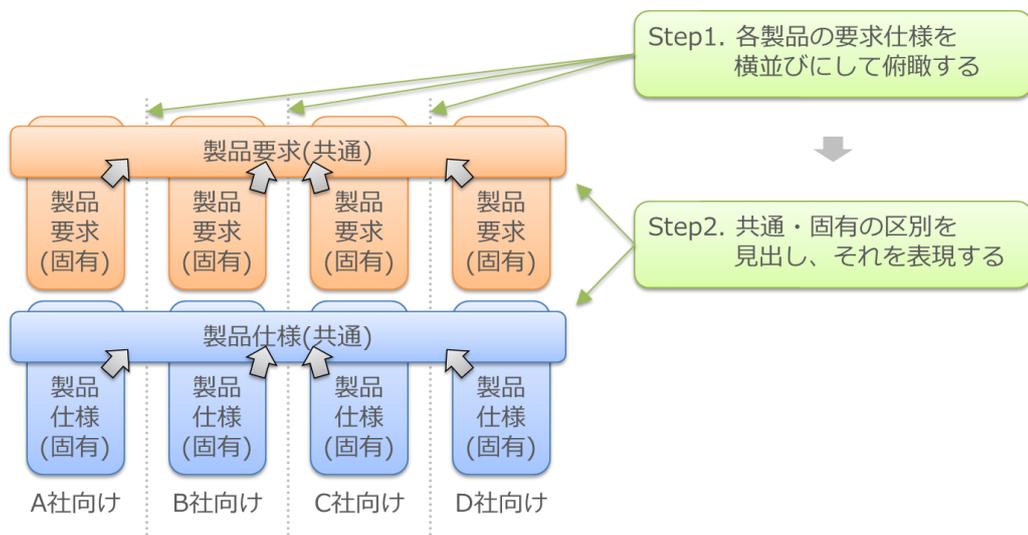


図 66-12 共通・固有の区別と定義

ここで、Step1 では USDM を用いて横並びさせることが可能であるが、それにより Step2 で見出す共通・固有の表現をどのように行うかという点が課題として生じた。ソフトウェアプロダクトラインの考え方においては、ツリー状のフィーチャモデルを用いて整理する方法が一般的ではあるが、今回の活動で扱う可変性はその粒度に相応しいのかどうかという懸念がある。たとえばソフトウェアでは表 66-2 のようなものが可変性として存在し、この単位の情報をツリーで表現すると図 66-13 フィーチャツリーの例のようになる。これでは管理対象の粒度が細かすぎ、ツリーの規模が大きくなりすぎるのではないかと懸念された。

表 66-2 扱いたい可変性の例：車両が走行中かどうかの車速による判断

A社向け製品	B社向け製品	C社向け製品	D社向け製品
XX[km/h]以上	YY[km/h]以上	ZZ[km/h]以上 +AA[ms]継続	出荷時設定の 閾値以上

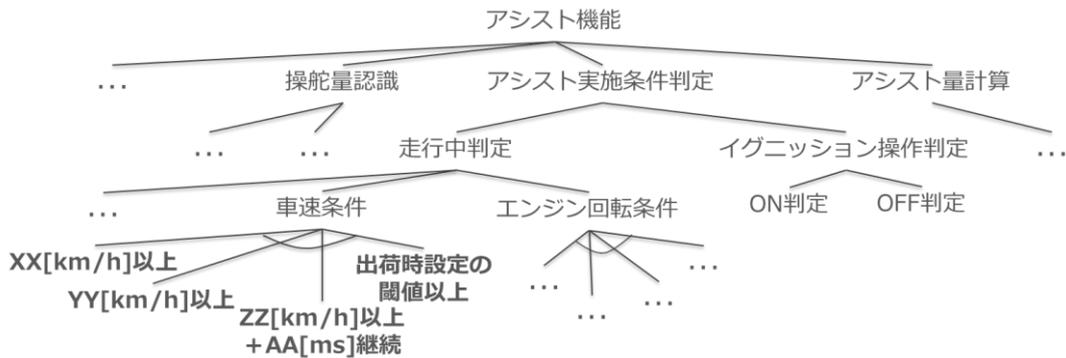
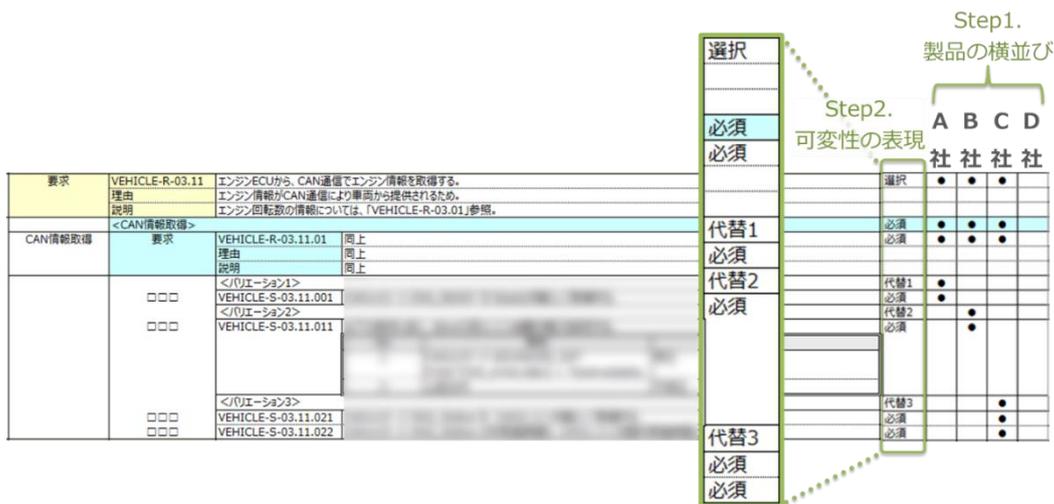


表 66-2 の項目は、「車速条件」として用いられる。
 車速条件は「走行中判定」に用いられるが、他にもエンジン回転条件（エンジンが回転しているか、止まっているか、不明かなどの状態）などの条件が存在する。
 走行中判定は「アシスト実施条件判定」に用いられるが、他にもイグニッション操作判定（ドライバーがイグニッションスイッチを ON しているか、OFF しているか）も考慮する。
 アシスト実施条件判定は最終的に「アシスト機能」の一環であるが、他にも操舵量認識（ドライバーがどの程度ステアリング操作をしているか）、アシスト量計算（どの程度のアシスト量を与えることにするか）といった機能が必要となる。
 このように、可変性として扱いたい表 66-2 の項目 1 つを取っただけでもこれだけのツリー構造が想像され、この手法でシステム全体分を表現しようとする、相当な規模になることが想定される。

図 66-13 フィーチャツリーの例

そこで今回の取り組みでは、USDM の記載を拡張する方法で表現することを検討した。その様子を図 66-14 USDM を用いた可変性の表現例に示す。右端の●印の列が Step1 で行う製品ごとの横並びであり、ボトムアップでまとめられた要求仕様の記載がそれぞれの製品に対応するのかが印付ける。これを俯瞰すると、全製品に共通するものなのか、ある製品に固有のものなのかという可変性が見出せる。これを Step2 の結果とし、可変性を表現する列を追加して記載することとする。



可変性の表現を列として追加した例。
 先頭行の要求は D 社製品では含まれないため、「選択」要素として扱う。
 第 2 階層の要求は、第 1 階層の要求が選択された上ではどの製品にも含まれているので、「必須」要素とする。
 この要求を実現するための具体的な仕様が、各社製品で異なるとする。するとこれは「代替」要素として扱われる。

図 66-14 USDM を用いた可変性の表現例

この取り組みにより、以下の効果が得られた。

- 共通と固有を見出し、それを USDM の拡張で表現する方法を確立した。
- これにより、標準成果物としてカバーすべき範囲の検討に有益な情報をまとめることができた。

5. 取り組み事例：PFD を用いた開発プロセス構築

5.1 上位階層の開発プロセス

課題③：上位階層の開発プロセスについては、「4.1.2」でトップダウンアプローチを実施する際に、その実施方法をプロセスとして明確にすることを試みた。

トップダウンで要求仕様を USDM としてまとめていく際は、アプローチの中で述べたように、各階層のアーキテクチャも見出していく必要がある。また冒頭で述べたように、機能安全規格対応が求められる開発部位（E/E システム以下）が存在するため、そのこともプロセス上考慮していく必要がある。これらを加味して開発プロセスを検討する。

図 66-15 に検討の様子を示す。まず今回検討している各階層の構成要素を、下階層へ繋がる成果物を出力するプロセスとして捉えることで、全体を PFD で表現する（図 66-15 左）。PFD ではプロセスを二重丸表記することで階層表現できるので、各構成要素のプロセスはそちらで詳細に検討する（図 66-15 中／右）。

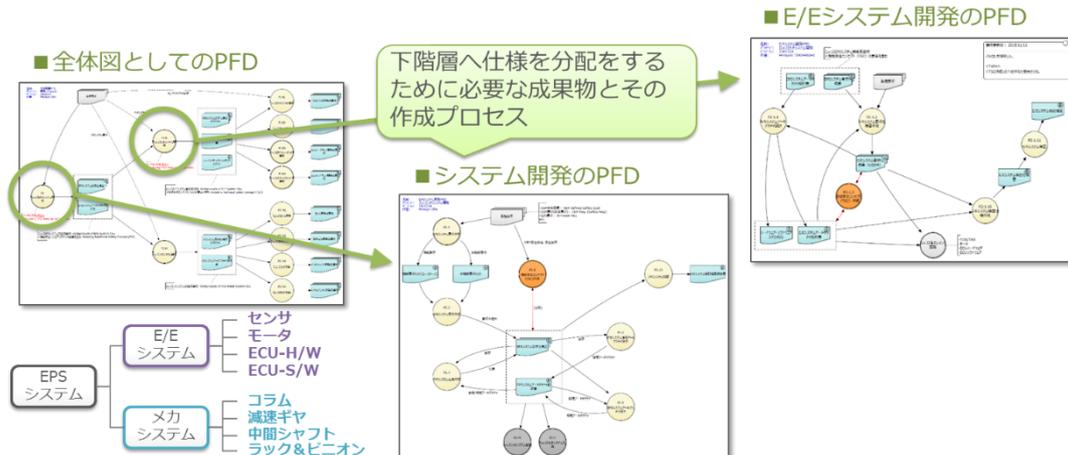


図 66-15 上位階層の開発プロセス検討

このように開発プロセスを PFD で可視化することで、以下のような効果が得られた。

- 業務内容の議論の活発化につながった。
「この成果物はここまで書ければ下階層は困らない」、「このプロセスは成果物が多すぎるので分割すべき」、「機能安全対応はこの段階で行うべき」、などの議論が、プロセスと成果物の連鎖表現により発想しやすい。

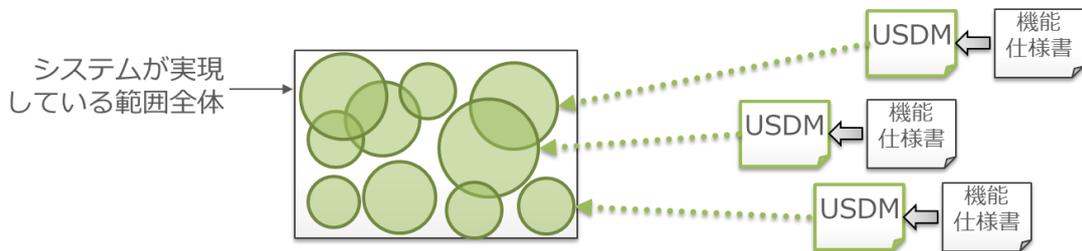
6. 考察および今後の取り組み

考察として、手法を適用したことで得られた気づき、および今後の取り組みを述べる。

6.1 手法適用における気づき

6.1.1. USDM：要求の獲得には別手法が必要

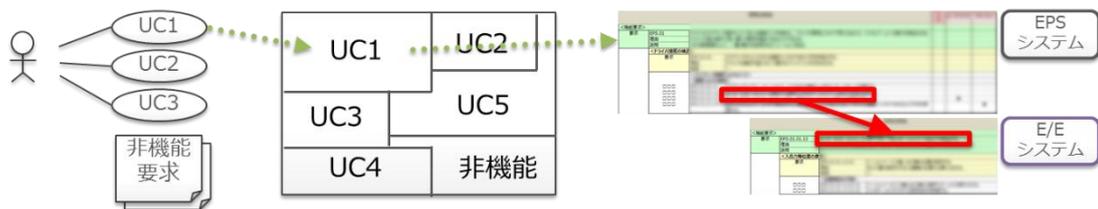
「4.取り組み事例：USDMを用いた要求仕様書構成」ではUSDMを用いて要求仕様を整理する様子を述べた。ボトムアップアプローチにより既存成果物の「仕様」部分から要求および理由をまとめていくが、それだけでは漏れなく重複のない要求仕様書が得られる確信は持てない。この様子を図 66-17 に示す。



1つの機能仕様書からボトムアップで表現できるUSDMの範囲は、システムで実現している範囲全体のうちの一部分をカバーする。これを全部集めたとしても全体が埋められるわけではなく、また重複する部分も見られる。

図 66-17 ボトムアップアプローチから得られる要求仕様書の範囲

そこで、トップダウンアプローチを用い上位階層から要求を明確にして行く。最上位となるEPSシステムへの要求はどこから来るかと考えると、それはやはりユースケース分析などの手法を用いて見出す必要がある。これによりシステムが実現すべき範囲を漏れなく重複なく明確にした上で、最上位要求として展開していくというアプローチが必要となる。この様子を図 66-18 に示す。



ユースケース分析などを用い、システムが実現する範囲全体を現す要求を漏れなく重複なく明確にする。その上で階層的に要求仕様書に落とし込んでいく。

図 66-18 最上位の要求の獲得には別手法が必要

6.1.2. USDM：自由と規律のバランスの良さ

さらに、「4.取り組み事例：USDMを用いた要求仕様書構成」においては、USDMフォーマットを用いながら要求仕様とアーキテクチャを対応付けることや、プロダクトライン開発を見据えた可変性の表現を実現することもできた。これはUSDMの持つ「自由と規律のバランスの良さ」が貢献していると考えられる。

USDMはその名（Universal Specification Describing Manner）が示すように、記述の際のマナーを示すものである。マナーとして、例えば要求・仕様・理由を階層化して表現することや、要求や仕様の分割基準を示すこと、また仕様化は要求の中の動詞を見つけて行うことなどを示している。こういったマナーを守ることで、USDMが本来目指す漏れなく・重複なく・関係者が共通認識を持てる要求仕様がまとめられるようになる。そのマナーを守った上で、例えばExcelという広く使われているツールで構成されるという点を利用し、「4.2」で述べた可変性の表現など、自分たちの開発に適したカスタマイズが行えるようになる。

6.1.3. PFD：プロセスと成果物の「連鎖」表現が効果的

「5.取り組み事例：PFDを用いた開発プロセス構築」では、PFDを用いてプロセスを可視化する取り組みを行った。この取り組みを通じて、PFDの特徴であるプロセスと成果物の連鎖という点に大きな意義を見出すことができた。

プロセスを検討する上では、過去の経験などから「どんなことをするか」、「どんなものを作るか」という想像がある程度浮かんでくる。それを表のような形でまとめることも可能である。実際に筆者の組織でも過去に用意されたことがある。しかしこのような形式だとプロセスのイメージが把握しにくく、また入出力成果物が全プロセスでつながっているのかどうかということの確認が困難である。この様子を図66-19に示す。

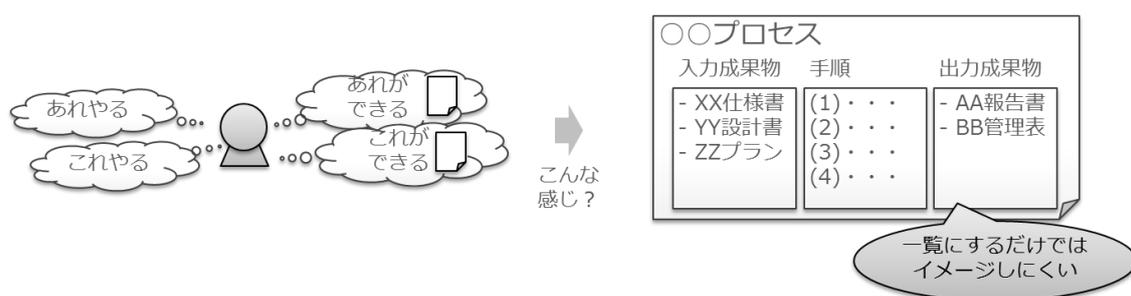


図 66-19 プロセスを表形式などで記述すること自体は可能

これをダイアグラムという形で表現することで、発想が整理されていく様子を実感できた。従来の表形式でまとめていたものと同じように、ある手順は何かを入力して何かを出力するものとして定義される。プロセスと成果物の連鎖がまさにそのことを指しており、そのダイアグラム表現が非常に直感的であることから、発想の整理がしやすくなっていると感じられた。最終成果物を思い描いたとき、それに至るには中間成果物として何が必要か、そのためには何をどれだけ検討すれば良いか、という思考が非常にスムーズに整理できることが実感できた。

6.2 現在位置とこれからの対応

現在は、「4.取り組み事例：USDM を用いた要求仕様書構成」および、「5.取り組み事例：PFD を用いた開発プロセス構築」で紹介した取り組みを通じて標準成果物の作成を行っている状況である。USDM を用いて暗黙知を形式知に明文化する作業を進めており、それは PFD で可視化したプロセスに沿って実行することで、そのプロセス自身の検証も行っている。

今後は標準成果物を完成させ、それを基にした派生開発を実際に回すとともに、標準成果物や開発プロセスの妥当性を検証し、ブラッシュアップを図っていく。

参考文献

- [1] 清水吉男：【改訂第2版】[入門+実践] 要求を仕様化する技術・表現する技術～仕様が書けていますか？、株式会社技術評論社、2010
- [2] 清水吉男：「派生開発」を成功させるプロセス改善の技術と極意、株式会社技術評論社、2007
- [3] 清水吉男：PFD (Process Flow Diagram) の書き方 第3版、株式会社システムクリエイツ、<http://soft-koha-hp.la.coocan.jp/process/PFDform3.PDF>、2009

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)