

ETSS-JMAAB解説書

2015 年度版

Version 1.0

■ 著作権について

本ドキュメントの著作権は、JMAABに帰属します。

JMAABは、本文書の内容に関し、いかなる保証もするものではありません。万一本文書を利用して不具合等があった場合でも、JMAABは一切責任を負いかねます。また、本文書に記載されている事項は予告なしに変更または廃止されることがありますので、あらかじめご了承ください。

■ 本ドキュメントの取扱いについて

本文書は、非営利目的、または利用者内部で使用する場合に限り、複製が可能です。また、本文書を引用する場合は、本文書からの引用であることを明示し、引用された著作物の題号や著作者名を明示する等の引用の要件を満たす必要があります。

本成果物については、JMAABホームページ(<http://jmaab.mathworks.jp/>)を参照ください。

その他のお問い合わせは、JMAAB事務局 (jmaab-office@mathworks.co.jp)へご連絡ください。

ETSS-JMAAB 変更履歴

| Revision | 日付 | 内容 | 作成者 |
|----------|----------|-------------------------|-----------------|
| 0.0 | 06/06/29 | MBDエンジニアスキル標準ドラフト版 | 育成 WG#1 |
| 0.1 | 07/03/12 | MBDエンジニアスキル標準ドラフト改訂版 | 育成 WG#2 |
| 1.0 | 07/06/11 | MBDエンジニアスキル標準初版 | 育成 WG#2 |
| 1.1 | 08/07/31 | ETSS 2007 準拠 | 育成 WG#3 |
| 1.1.1 | 09/10/8 | JMAAB 事務局移管にともなう連絡先等の変更 | MathWorks Japan |

ETSS-JMAAB 解説書 変更履歴

| Revision | 日付 | 内容 | 作成者 |
|----------------|----------|--|--------------------------------|
| 2015 Ver1.0 | 15/06/04 | ETSS - JMAAB 4 冊を 1 冊にまとめ解説書として 発行 自動車制御システム開発向けに全面見直した。 | ETSS - JMA AB フォローアッ プWG |

目次

| | |
|-------------------------------|----|
| I. はじめに | 6 |
| 1. 本書について | 6 |
| 1-1. 本書利用の最も重要な注意点 | 6 |
| 1-2. ETSS-JMAAB作成の背景 | 6 |
| 1-3. 本書作成の背景 | 7 |
| 1-4. 実践の注意点 | 8 |
| 1-5. メンバー | 8 |
| 1-6. 用語 | 8 |
| 2. 自動車制御システム開発 | 8 |
| 3. MBD | 11 |
| 4. MBDエンジニア | 11 |
| 5. MBDプロセス | 12 |
| II. ETSS-JMAABの概要 | 22 |
| 1. 組込みスキル標準（ETSS）との関係 | 22 |
| 2. ETSS-JMAABとは | 22 |
| 2-1. ETSS-JMAABの概要 | 22 |
| 2-2. ETSS-JMAABの構成 | 22 |
| 2-3. スキルレベルとキャリアレベルの考え方 | 23 |
| 2-4. ETSS-JMAABに期待される効果 | 25 |
| III. スキル基準 | 28 |
| 1. ETSS-JMAABスキル基準の概要 | 28 |
| 2. ETSS-JMAABスキルフレームワーク | 28 |
| 2-1. スキルの定義 | 28 |
| 2-2. ETSS-JMAABスキルフレームワークの構造 | 29 |
| 2-3. ETSS-JMAABスキルカテゴリの説明 | 30 |
| 2-4. ETSS-JMAABスキルレベルの説明 | 31 |
| 3. ETSS-JMAABスキル基準 | 33 |
| 3-1. 技術要素スキルカテゴリ | 33 |
| 3-2. 開発スキルカテゴリ | 34 |
| 3-3. 管理スキルカテゴリ | 35 |
| IV. キャリア基準 | 36 |
| 1. ETSS-JMAABキャリア基準とは | 36 |
| 1-1. ETSS-JMAABキャリア基準の概要 | 36 |
| 1-2. ETSS-JMAABキャリア基準の必要性 | 36 |
| 1-3. ETSS-JMAABキャリア基準の期待される効果 | 36 |
| 2. キャリア・フレームワーク | 37 |
| 2-1. 職種の区分 | 37 |
| 2-2. キャリアレベルの定義 | 37 |

| | |
|---|----|
| 2-3. 職種とスキルとの対応付け | 38 |
| 2-4. 職種の活動領域 | 38 |
| 3. キャリア基準 | 39 |
| 3-1. 職種とキャリアレベル | 39 |
| 3-2. 職種のスキルレベルとの対応付け | 39 |
| 3-3. 職種と責任及び役割の対応 | 40 |
| 3-4. 職種 | 40 |
| V. 付録 | 45 |
| 1. スキル定義例 | 45 |
| 1-1. 制御システムエンジニアスキル定義例 | 45 |
| 1-2. プラントモデルスキル定義例 | 46 |
| 1-3. 実装モデルエンジニアスキル定義例 | 47 |
| 1-4. 適合エンジニアスキル定義例 | 47 |
| 1-5. 開発環境エンジニアスキル定義例 | 48 |
| 2. 教育用スキル分解例 | 49 |
| 2-1. 制御システムエンジニアスキル分解例 | 49 |
| 2-2. プラントモデルスキル分解例 | 49 |
| 2-3. 実装モデルエンジニアスキル分解例 | 49 |
| 2-4. 適合エンジニアスキル分解例 | 49 |
| 2-5. 開発環境エンジニアスキル分解例 | 49 |
| 3. 活用事例 | 50 |
| 3-1. 開発環境エンジニアのための Simulink A P I スキル定義 | 50 |
| 3-2. W社推進プロセス事例 | 51 |
| 3-3. X社推進プロセス事例 | 52 |
| 3-4. Y社推進プロセス事例 | 54 |
| 3-5. Z社推進プロセス事例 | 55 |
| 4. 教育に関する注意点 | 56 |
| 5. プロジェクトで活用する際の注意点 | 58 |
| 6. 用語解説 | 60 |
| 6-1. スキル、技術 | 60 |
| 6-2. 技術者と技能者 | 61 |
| 6-3. スキル分布 | 62 |
| 6-4. キャリア基準 | 62 |
| 6-5. キャリア | 62 |
| 6-6. 職種 | 62 |
| 6-7. 責任 | 62 |
| 7. 教育プログラム | 63 |
| 7-1. 教育プログラムフレームワーク | 63 |
| 8. MBD用語 | 65 |
| VI. 図表一覧 | 68 |

I. はじめに

本章では、本書制定の背景、自動車制御システムやMBD等について説明する。

1. 本書について

1-1. 本書利用の最も重要な注意点

本書は、サンプルに添付されたスキルマップやキャリア基準をベースにして、各社が追加編集する為の物ではなく、各社が自社にあったスキルマップやキャリア基準を作成する為の手法をまとめたものである。添付資料はあくまでも参考に留め、そのまま使用しないように注意いただきたい。

1-2. ETSS－JMAAB作成の背景

自動車は、常に改良され続けているが、近年は、特に環境技術、安全技術における革新が強く求められている。また、従来にもまして利便性、快適性の改善も求められている。そして、これらの要求に応えるために、電子制御による高機能化、多機能化が急速に進んでいる。高機能化、多機能化によりソフトウェア（以下「S/W」と略称）が量的に拡大しており、開発工数が増大する一方、品質保証も大きな問題となっている。

自動車業界において、制御系の開発効率向上及び品質向上のためにモデルベース開発（Model Based Development、以下「MBD」と略称）への移行が不可欠となり、MBD実現のための必須ツールである MATLAB®/Simulink® が広く使われるようになってきている。しかし、現状では MATLAB/Simulink の機能が十分ではなく、効率化できていない制御仕様開発プロセスが存在していることも事実である。

Japan MBD Automotive Advisory Board（以下「JMAAB」と略称）は、日本の自動車業界の、MBDの推進と、MATLAB/Simulink ベースの設計・開発環境をより発展させるために、発起人を中心として設立された MATLAB のユーザ会である。

MBDの推進にあたり、ツールの機能改善とともに重要となるのがエンジニアの育成であり、この推進のために『JMAAB MBDエンジニア育成WG』というワーキンググループを発足した。2008年発行のスキル標準ETSS－JMAABは、『JMAAB MBDエンジニア育成WG』にてWGメンバーの各社で合意した内容を編集したものであり、自動車用制御装置の開発をモデルベースで進めていくエンジニア育成の指針としていくものである。

1-3. 本書作成の背景

E T S S - J M A A B の発行から数年が経過した時に、実際に活用する場合において、人材評価として使用できるが、教育に使用するとそのままでは使えないことが解ってきた。また、以前の成果物には、各社で活用するためスキル基準・キャリア基準を掲載していたが、その作成方法が明確ではなかったために各社で適切な運用ができていないことが見えてきた。そこで、新たに『E T S S - J M A A B フォローアップWG』を設立し、4年の時間をかけ、内容の見直しを行い新たに解説書として発行するに至った。特に、今回はスキルの抽出方法についてさらに詳しい解説を行い、各社が適切にスキルマップを作製できるよう注力している。

世間からのニーズとして、JMAAB にて標準的なスキルマップを作って欲しいと言う要望があることは理解しているが、そもそも各社が同じ製品を、同じように作っているわけではない。実社会では、各社の戦略の元、それぞれの会社が独自のプロセスを採用し、独自の技術を磨き競争している。

したがって、まず技術要素スキルは各社で全く異なるはずである。開発技術については、分野を限定的にすることによって、いくつかのスキルは標準化として定義できる。しかし標準として定義できるスキル（例えば、Microsoft® Word®）は単純な使い方であって、ほとんどの技能はドメインとの結びつきがあって初めて習得が可能となる。例えば Microsoft Word で表の作り方を習ったとしても、どのような場面で、どのように使用するのか解っていなければスキルを使えるとは言えない。表にまとめる技術は小学校でも習った算数である。その当時から現段階までに、さまざまな事例を表にまとめる問題を習ったはずであり、その結果、皆、苦勞せずさまざまな問題を直感的に表にまとめることができるのである。単純なスキルであっても自動車業界という狭い範囲に対してさえ、標準化は難しい。

Simulink の習得スキルというものを考えてみよう。「自動車業界 MBD エンジニアのための Simulink 入門 (TechShare 出版)」の第 2 章に簡易なスキルマップが掲載されている。しかしながらそれが、自動車業界全体で共通的に使えるわけではない。レベル 1 程度の本当の入り口部分は、ほぼ一致したスキル定義になるであろう。しかし、各社使用する Simulink ブロックの種類や、構造化の設計手法は異なる。日本の自動車メーカー全社が AUTOSAR に準拠した S/W を設計しているわけではない。各社各様、したがって、当然、スキルはレベルが上がるほど互換性はなくなる。

つまり重要となるのは、スキルマップやキャリア基準の作り方である。

本書は、そのために参考となるスキルマップを掲載し、「解説書」という位置づけである。あくまでも作り方の参考であり、内容をまねしてはいけない。やり方、手法をまねすべきである。そして、実のある成果に結びつく使えるスキルマップやキャリア基準を作成していただきたい。本書は、それを目的に編集を行った。

1-4. 実践の注意点

この解説書は、スキルマップを作ることが最終目的ではない。対象とするエンジニアを各社が求める人材として育成することが目的である。本手法による人材育成は、正しく運用しなければ、工数が増えるだけ、あるいは技術者のモチベーションを下げってしまう「もろ刃の剣」となりえる。運用する側も技術者目線で正しく実践すべきである。

1-5. メンバー

◆ JMAAB MBDエンジニア育成WGメンバー：

アイシン精機株式会社、アイシン・エイ・ダブリュ株式会社、株式会社アドヴィックス、いすゞ自動車株式会社、スズキ株式会社、株式会社デンソー、トヨタ自動車株式会社、日産自動車株式会社、株式会社日立製作所、株式会社本田技術研究所、マツダ株式会社、三菱電機株式会社、ヤマハ発動機株式会社、ジヤトコ株式会社、日産ディーゼル工業株式会社（現UDトラック株式会社）、株式会社ミツバ（順不同）

◆ E T S S - J M A A B フォローアップWGメンバー：

アイシン・エイ・ダブリュ株式会社、株式会社アドヴィックス、株式会社デンソー、トヨタ自動車株式会社、日立オートモティブシステムズ株式会社、富士通テン株式会社、株式会社本田技術研究所、三菱電機株式会社、ヤマハ発動機株式会社、UDトラック株式会社、株式会社ミツバ（順不同）

1-6. 用語

本ドキュメント特有の用語は、付録に記載する。それ以外の用語については、『JMAAB MBD用語集』と『Japan Calibration User Group 適合用語集』を参照して頂きたい。

2. 自動車制御システム開発

自動車は大量生産工業製品であり、制御システムは技術者の手を離れてさまざまな環境と目的で利用される。制御システムはS/Wだけでは成り立たず、制御S/Wと物理システムが一体となって機能を発揮する。制御S/Wは制御対象（以下「プラント」と略称）の物理特性に応じて数ミリ秒ごとにアクチュエータ操作を時系列で決定しなければならず、その決定したアクチュエータ操作によりプラントの振る舞いに変化する。決定が不適切であれば、プラントの振る舞いはどんどん望ましい状態から遠ざかってしまうため、プラントの制御開発者の願望だけでアクチュエータの振る舞いを決めることはできない。

このようなシステムの開発要求と制約を事前に明らかにすることは容易ではなく、開発中に試作と実験を繰り返すことで明確になってくるものである。例えば、「燃費を10%向上」が1つのシステム開発要求だったとしても、そのために実現しなければならない要素開発項目は無数にあり、相互依存や背反も複雑に絡み合っている。これらの開発項目の関係を実験やシミュレーションなどを駆使して明確にし、適切な開発戦略を決めていくことはシステム開発そのものであり、開発は要求と制約を明確にする過程であるということができる。したがって、常に、明示された要求と制約は見落としや間違いがあることを前提として、開発された自動車が十分な安全性・信頼性と設計性能を発揮するかを可能な限り全ての運転条件で調べ尽くすことが自動車制御システム開発の基本である。すなわち、S/Wといえど試作と検証を繰り返しながら改善を行うものであって、S/W要求と制約を絶対正とするものではないのである。

さらにいえば、自動車業界は車の安全性・信頼性・性能などの市場実績に基づき継続的な改善を実施することによって、高安全・高信頼・高性能な車を合理的な価格でユーザに提供するシステムを構築してきたのである。市場実績が重要である理由は、入念な検証を積み重ねて市販しても、仮に何らかの不具合があれば、非常に多くのユーザによって多様な環境で運転されるのでたちまち顕在化してしまうからである。すなわち、開発中の運転条件はいくら入念に実験を計画しても市場で曝される運転条件を超えることができないのである。

自動車業界では自動車メーカーとサプライヤとの密接な連携によって自動車の開発が行われる。自動車メーカーの開発パラダイムは前述した通りであるが、S/Wサプライヤは基本的には自動車メーカーからの開発依頼に従って、制御S/Wの一部、あるいは、開発部品に対するECUと制御S/Wを開発する。Tier1 サプライヤはS/W会社などにさらに一部の開発を委託することもある。そのため、自動車メーカーや開発のより上流の会社から下流の会社への仕様提示が絶対正のものと考えられ易い。しかし、上流の会社の仕様は往々にして不完全であり、やり直しが発生することが多い。S/W開発の下流側からは上流に対して仕様精度をあげることを要求したくなることはごく自然である。しかしながら、前述の状況からこの要求に答えるには限界がある。むしろ、繰り返しは開発プロセスの一部となっているので、要求精度を向上させるよりも、繰り返しを効率化することが本質である。とはいえ、あまりにもレベルの低い上流側からの提示は責められるべきである。自動車メーカーとサプライヤが制御システム開発は改善によって、さらにいえば、世代を超えた継続的改善によって行われるものであるという認識を共有し、適切な連携関係を構築しなければならない。これが、JMAABの合言葉の一つの「自動車メーカーとサプライヤ間の連携による効率的な開発環境」の意味である。もし、S/Wサプライヤが自動車メーカー要求の通り確実にS/Wを開発し、その結果制御システムとして不具合が生じた場合、S/Wサプライヤが「仕様通りである」と自身の正当性を主張し、自動車メーカーがそれを認めたとしても、そのS/Wサプライヤには2度と発注が行かないだろう。自動車メーカーは常に自動車開発全体の効率化を望んでいるので、狭い視野で強行に自身の正当性を主張し、全体の開発効率化を阻害するサプライヤは不

都合な存在に見えるのである。実際、Tier1 サプライヤは、自動車メーカーの開発の状況を理解できる人材育成に力を入れている。

一方、自動車業界は非常に厳しい競争に曝されていて、開発期間の短縮・効率の向上が常に求められており、「開発に先立ち要求を定義し、それを確実に実現していく。」というパラダイムでは不十分となってきている。現実的には、要求を明確にするという作業は極めて難しく、前述したように、要求と制約を明確にすることが自動車制御システムの開発そのものである。つまり、車体やエンジン・駆動系のハードウェア（以下H/Wと略称）を試作し、同時に制御S/Wの試作を素早く開発し、膨大な実験と評価に基づき改善を進めるのである。改善の1つのループ中に実施される実験と評価は膨大であり、自動車H/Wと制御S/Wを同時に開発し、不具合が見つければ直ちに修正する素早い改善を行わなければ合理的な時間で開発を終えることはできない。制御システムの急速な複雑さの進展により、従来型の開発では指数関数的に増大する評価項目や必要な実験に対応することは難しくなってきた。このため、初期のH/Wと制御S/Wの試作品質を向上させるために、制御S/Wを加えた閉ループ系での自動車やエンジン・駆動系などの振る舞いをシミュレーションできるモデルを利用すること、開発中にバーチャル環境とバーチャルとアクチュアルを組み合わせた環境、アクチュアルな環境を適宜利用することで素早い繰り返しによる改善を行うMBDが指向されるようになった。

すなわち、MBDは継続的改善を素早く実現するための手段であり、その本質は開発中に可能な限り自動車が市場で遭遇する運転条件を評価し尽くすことである。バーチャル環境での実験と評価によって、初期の試作品質を向上するとともに、開発中に見出された不具合は直ちに修正し、それぞれの工程で不具合が下流工程に流れることを防止する。そのためにモデルを用いた高精度な検証を用いるのである。また、各開発工程での高精度な検証は工程自体の品質を健在化させるので、その工程の改善も効率的に行うことができる。

前述のような自動車開発の世代を超えた継続的改善が自動車制御システム開発の全貌であり、「開発に先立ち要求を定義し、それを確実に実現していく。」というパラダイムは開発の一部に適用できるに過ぎない。それを開発全体に適用しようとすることは無理があり、本質的に誤りである。一方で、「開発に先立ち要求を定義し、それを確実に実現していく。」という努力を怠れば、開発の効率化は望めない。できる限り「開発に先立ち要求を定義する。」という努力は行うべきである。また、世代を超えて要求と制約を蓄積し、より精度の高い開発要求を開発に先立って提示する仕組みを構築することに努力を払うべきである。実際、自動車業界には自動車メーカーとサプライヤにその蓄積がある。しかしながら、「全ての要求を定義する。」ということが行き過ぎると、実現不可能な過剰な要求ベース開発に陥り、結果に繋がらない不合理な負担を開発者に強いることになり、リソースを無駄なところを使うことになる。その結果、非常に高コストな開発になり、高安全・高品質・高性能な車を合理的な価格でユーザに提供することが阻害されてしまう。時には、高安全・高品質・高性能が

損なわれることもあるだろう。予算・人材などのリソースには限界があることを忘れてはならない。

3. MBD

複数化・高度化した現代の自動車制御システム開発において、MATLAB/Simulink等のCAEツールによって制御装置と制御対象の機能をモデル化し、開発の全プロセスにおいて、シミュレーション技術を活用することで、製品ライフサイクル全般にわたった品質向上と開発効率向上を目指した開発のことである。

モデルの定義：コンピュータが理解できる対象の簡略表現

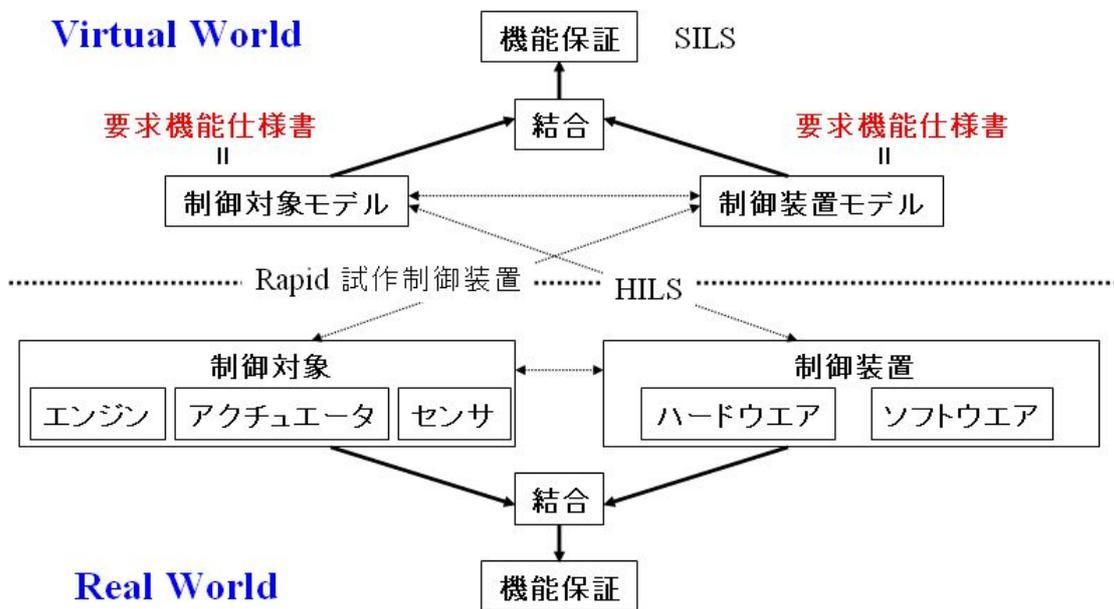


図 1. 制御システムにおけるモデルの位置づけ

4. MBDエンジニア

一般的にエンジニアとは、日本語では技術者と解釈される。主に工学（エンジニアリング）分野の専門的な技術を持った実践者のことであり、製造業一般、サービス産業など製品やシステムなど、モノやことを生み出される、生産がともなう全ての産業に存在している。即ち、何らかのものを作り出す人のことであり、昨今では製作したものを正しく動作させるための運用・保守に関わる職種も含まれている。一方、技術者に類似した概念に技能者がある。技能者とは、ものづくりの実作業を担当する者を指し、伝統的な職人の概念に近い。エンジニ

アは技能者としての一面も持つが、どのようなものを作るか、どのように作るか、どのように運用するかなどを考えることが重要である。

MBDエンジニアとは、MBDの専門的な知識と技術を持ち、製品の「開発」「改善」「革新」「実現」を行う実践者と言える。一方、製品開発はMBDの知識のみで行えるものではなく、製品に対する深い理解と知識も不可欠である。ETSS-JMAABにおいては、この両方の知識と技術を兼ね備え、自ら考える力を持って製品開発の職務遂行を実施する技術者を、MBDエンジニアと呼ぶ。

5. MBDプロセス

JMAABではモデルベース開発の標準的な開発プロセスを定義し、そこで使われる開発スキルが各社の技術要素スキルで共通的に使われる開発スキルに対応すると考え、標準プロセスの定義を行い、製品の開発プロセスとして以下のようなプロセスを考えた。

5-1-1. システム開発プロセス全体

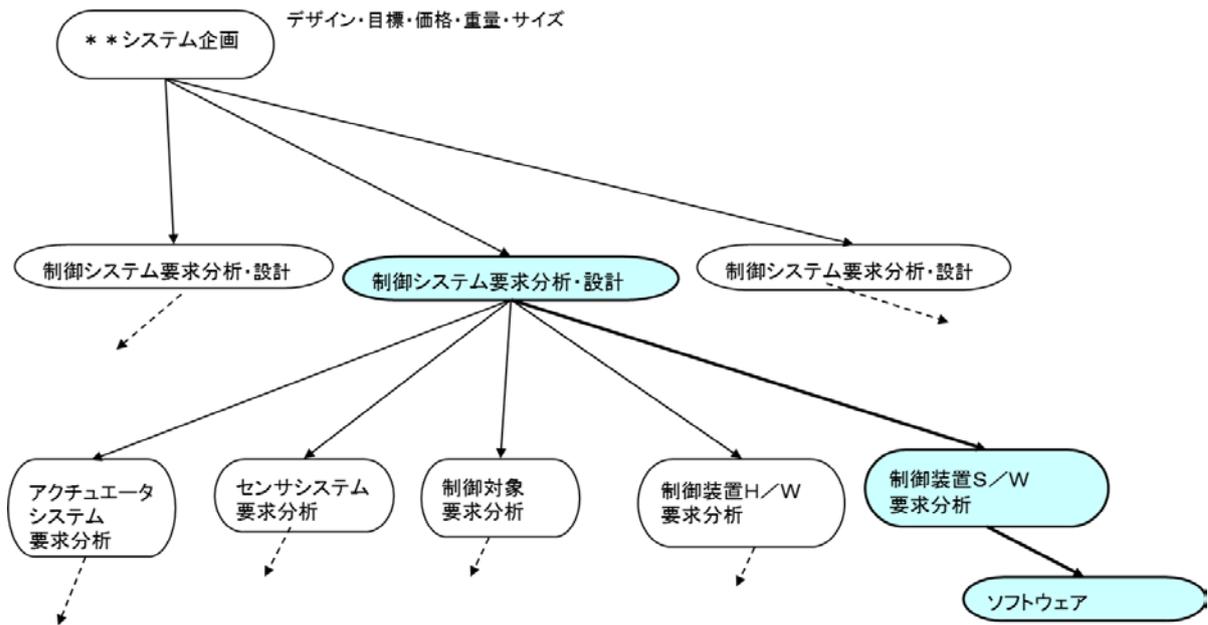


図 2. システム開発プロセス

単体でシステムという場合もあるが、一般的にシステムは複数のシステムで構成されることが多い。自動車を考えても、多くの部品で構成されており、その一つ一つがシステムと言えよう。

「図 2. システム開発プロセス」では、自動車というシステムには無数の制御システムが存在するということを表している。制御システムごとに、制御対象、制御装置H/W、制御

装置S/W、アクチュエータ、センサといったシステムで構成されている。そのそれぞれに要求分析から検証までのプロセスが存在している。

制御システムの要素ごとにプロセスは存在するが、ETSS-JMAABでは、システム企画以下のS/Wに関するプロセスについてのみを取り出してMBDプロセスとしている。大まかに企画、機能設計、実装、適合、制御システム検証、プロセス改善の6つのプロセスに分類している。

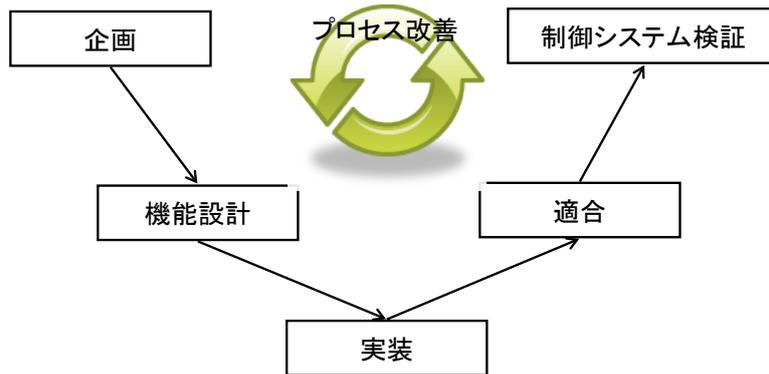


図 3. MBDプロセス

さらに、機能設計から制御システム検証の制御装置S/Wに関するプロセスについては、下図のようにV字プロセスとして詳細定義を行い、ドキュメントの流れを記述した。

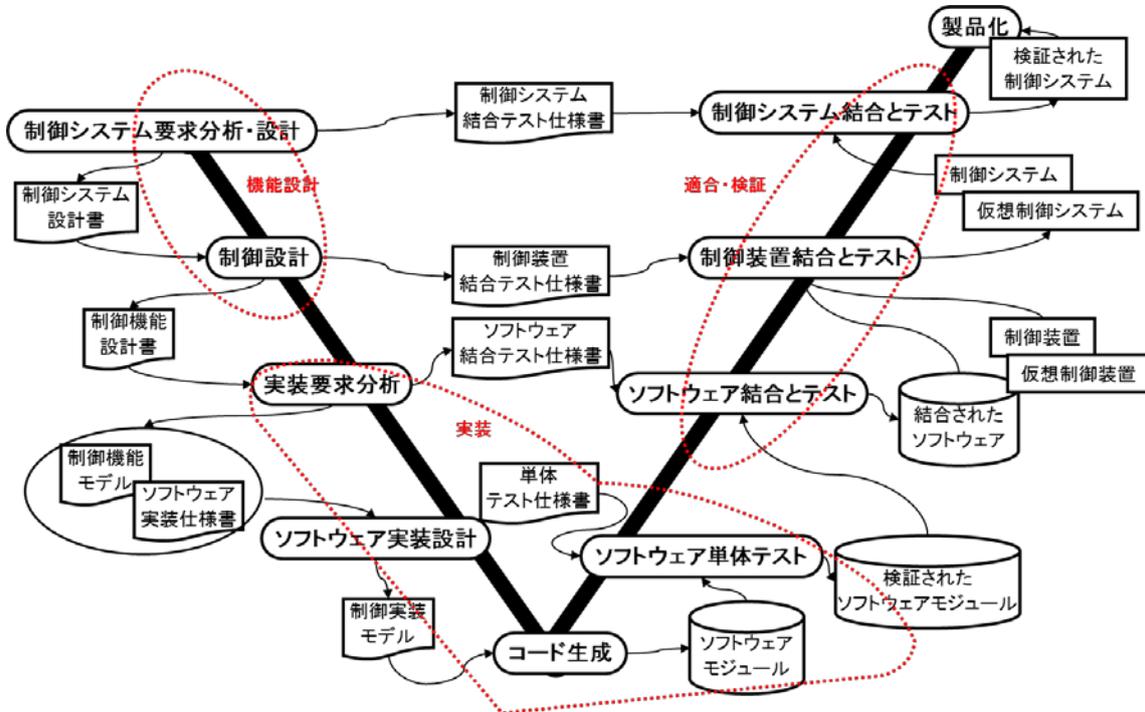


図 4. S/W開発プロセス

5-1-2. システム定義

今まで、何度もシステム、制御システムという言葉が出てきた。これらの言葉は何を指すのか？ここで、ETSS-JMAABで定義したシステムの内容について説明する。

下図は、システムと制御システムの関連を表す。すなわち、システムは複数のシステムから構成されており、その中には制御システムも含まれていることを示している。

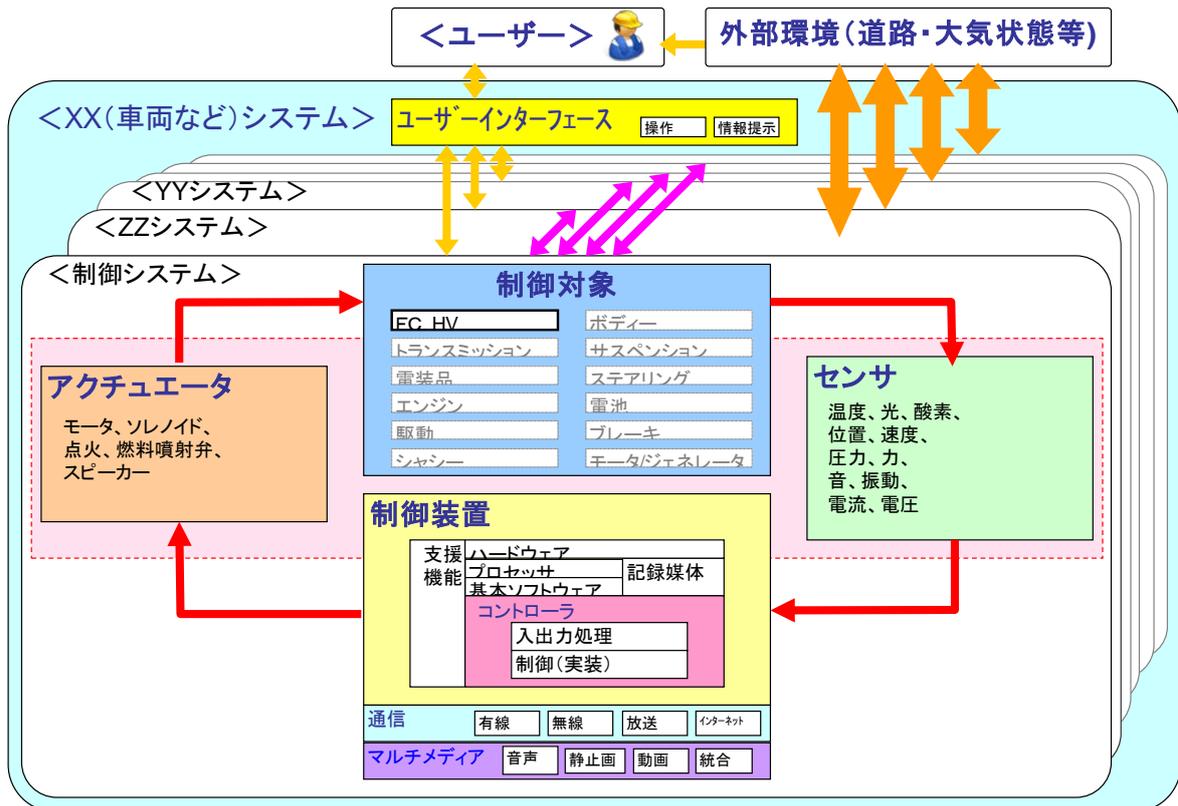


図 5. システムと制御システムの関連

ETSS-JMAABでは下記のとおり定義する。

■ システム

個々の要素が相互に影響しあいながら、全体として機能するまとまりや仕組みのことである。組み合わせのどの範囲を切り取っているのか、全体の中のどの階層に注目しているのか、そしてどの機能的特徴に注目しているのか、などによってさまざまなシステム名称を持つことになる。システムは以下の性質を持つ。

- システムはいくつかの要素によって構成されており、システムに含まれる全ての要素は、必ず自分以外の要素に対してなんらかの影響を及ぼす。
- システムは時間、または時間に写像できる順序集合（全順序集合）に沿って動作する。

- どのようなシステムでも、そのレベル内には複数の要素をもち、さらに異なるレベルの階層の積み重ねにより、機能を実現している。

注意：「システム」と表現するとイメージする対象は人によって必ず異なる。システムを使用する場合、「〇〇システム」と必ず対象を限定する表現を用いること。

■ 制御システム

他の機器やシステムを管理し制御するための機器、あるいは機器群

5-1-3. MBDプロセスの概要

参考として、「図 3. MBDプロセス」の各プロセスの概要を記載する。MBDプロセスはその1つ1つのプロセスをさらに分解すると、まだ小さなV字プロセスがあり、その中には要求分析、モデル化、検証といったプロセスが含まれている。MBDは検証を前倒ししフロントローディングを行えると言われるのは、それぞれのプロセスにおいて検証を行えることが所以である。

(1) 企画

自動車の製品企画は、商品構想、事業構想、技術構想を柱として構成され、開発の開始となる。

- ① 商品構想は、製品を投入する市場やユーザに関する事項を決定するものである。
(誰のために、どんな価値を提供するのかなど)
- ② 事業構想は、投資と利益のバランスを考え、事業として成立するか検証するものである。
- ③ 技術構想は、製品の機能が実現可能で安全に提供できることを担保するために、基本となる技術要件を決定するものである。

ここではMBD制御系開発に直接繋がっていく技術構想を中心に企画の説明をする。

企画時の技術構想として「(1)商品コンセプト」、「(2)プラットフォーム」、「(3)主要諸元とパワートレイン」、「(4)技術的特徴」、「(5)製品のポジショニング」、「(6)スタイリング」などが決められる。さらにここに事業構想としての「(7)コスト目標」、「(8)投入時期」が加えられ、仕向け地に関する制約条件として「(9)各種規制」、「(10)使用環境など」が考慮される。

自動車のコンポーネントの企画にブレークダウンすると、「(3)主要諸元とパワートレイン」の中のシステム概要として、基本構成、織込む技術を決め、搭載車両、仕向け地、性能、コストなどの目標、さらには仕向け地における制約条件として「(9)各種規制」、「(10)使用環境など」が提案され、シミュレーション等で提案内容の妥当性を確認して最終決定していく。

MBD制御系開発において、シミュレーションはモデルを活用して行うことが基本となる。モデル化のために必要な情報を抽出し、要求を分析し、モデルへの要件を決める。この時点

のモデルは実験モデル、物理モデルがあり、状況によって使い分ける。このモデルを使いパラメータスタディ等を行い妥当な目標値を決める。

基本的にこの時点では性能の目標が達成できそうかを判断できるが、コストの目標についてはモデルを使ったシミュレーションでは判断できない。よってコストに関しては、過去のデータをベースとして数値目標が決められる。

以上のような工程で企画内容が決められる。

(2) 機能設計

機能設計とは、上流工程である企画で決定された要求をより具体化し、次工程である実装可能レベルに詳細化することであり、「システム方式設計」「モデル化」「検証」の3つの工程からなる。

「システム方式設計」では、要求仕様を分析し要求を満足する中で、システムの制御対象（プラント）と制御装置（コントローラ：ECU）の機能分担を明確にする。要求仕様の分析では、振る舞いを定義する機能要件とコスト・信頼性・保守性などの非機能要件をそれぞれ抽出する。自動車の開発において、信頼性は重要な項目の一つであり、故障解析を実施し、要件より定まる各種の制約条件を満足するようにH/Wの構成、S/Wによる故障検出、故障判定時の振る舞いを考慮したシステムの構成を決定する。

「モデル化」では、制御対象・制御装置の夫々の振る舞いを定義する。機能設計上の制御対象のモデル化は制御装置の検証を行うためのものであり、検証の目的に応じて要求される精度が異なるため、精度に応じたモデル化が必要となる。また、HILSによりシミュレーションを行う時は、制約条件に応じたモデルの考慮が重要となる。

「検証」では、定義した振る舞いが正しく機能し要求を満たすことを確認する。ここでは、制御対象、制御装置、各々が設計上正しく動作するかを確認するためテストケース作成を行い、制御対象、制御装置を組み合わせ上位の要求を満足しているかの妥当性検証を行う。

(3) 実装

実装とは、ECUを動作させるために必要な制御S/Wをマイコンへ組込むことである。ECUを正しく動作させるS/Wの実装工程は、大きく「要求分析」、「プログラミング」、「検証」の3つに分類される。

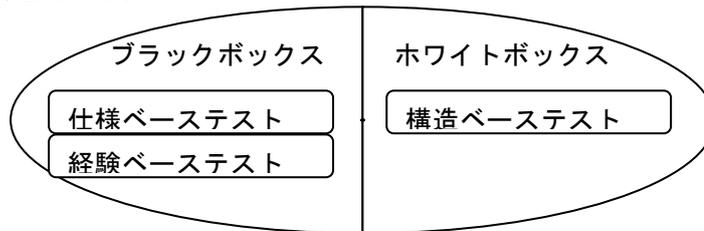
「要求分析」では、H/Wやマイコンの仕様・制約条件を正しく認識したうえで、制御モデルの『機能』の実現方法を決定する。

- ・ROM/RAM/スタック容量内でS/Wを搭載可能か判断する
- ・通信容量・速度などの制約条件下で、機能が正しく動作することを検証する
- ・モデルの階層構造を決定し、上記条件を満たす保守性の高いモデルで機能を実現する

「プログラミング」では、制御モデルを実装モデルへ変換し、Cソースを生成する。信頼性、安全性、移植性を考慮したコード効率のよい実装モデルの検討を行う。ここでは、モデルからの自動生成、手動生成かは問わない。生成されるCソースをイメージし、動作タイミ

ングの設定、コード特有の機能（ゼロ割防止、上下限設定、オーバーフローなど）を考慮する必要があり、マイコンやCソースに対する知識が要求される。個々の機能の単体テストはここに含まれ、生成したCソースに対して制御モデルで実現されていた機能の確認を実施する。実装モデル部分とモデルで実現できないCソース部分の結合を行い、実装のためのオブジェクトを生成する。ここでは、コンパイラやアセンブラに対する知識が要求される。

「検証」では、制御仕様に記述した機能、及び前提とする制約条件を満足していることを確認するテストを行う。各プロジェクトに最適なテスト内容を考え選定する。テスト選定で重要なことは、「どのテスト」で「何」をテストするかを常に意識して検証することである。検証は、大きく「仕様ベーステスト」、「構造ベーステスト」、「経験ベーステスト」の3つに分類される。「仕様ベーステスト」で機能/非機能/負荷について検証する。「構造ベーステスト」でデータフローパス/モデル静的検証・動的検証/Cソース静的検証・動的検証を行い、モデル、Cソース、オブジェクトについて等価性検証を実施する。「経験ベーステスト」は、過去不具合などから抽出した観点から行うテストで再発防止が期待できる。実施したテスト結果は、制御モデルとのトレーサビリティを取り、「どのテスト」で「何」を検証したのかを関連付ける。



(4) 適合

適合とは新型車両用に開発された制御仕様を、実際のコンポーネントや車両に組込んで実験を行い、制御定数を定めることである。この適合の結果が新型車両のECUプログラムに反映される。

以下にエンジン適合の例について説明する。

エンジン適合では、車両の商品性、性能、法規対応のため、要求性能であるエンジン性能、動力性能、燃費、排気、高温・低温性能、ドライバビリティ、始動性、各種燃料対応、各国排気規制対応、OBD規制対応等々をクリアし不具合のないレベルとするようエンジン及び車両システムを構築し、ECU/Wの定数を合わせこんでいる。

適合の手順を大まかに説明すると、「①エンジン試験」、「②車両試験」、「③実走行試験」、「④環境試験」の工程がある。

① エンジン試験

エンジン単体で全運転領域の各種データ計測を行い、車両では再現の難しい、幅広く複雑な運転状態におけるエンジンデータを包括的に計測し、膨大な制御パラメータの最適化を実施する。また、エンジン単体での出力性能や耐久性能なども同時に評価する。

② 車両試験

実走行を再現した状態で各種データ計測を行う。定速走行に限らず、加減速走行、登坂走行、高速走行といったさまざまな走行状態における車両データ計測を行い、要求走行性能への適合と最適化を実施する。また、燃費、排気ガスといった環境性能や音、振動といった官能評価なども実施する。

③ 実走行試験

テストコース等で運転時の官能評価を行う。エンジン試験、車両試験を通して最適化された車両で、実走行状態における音・振動・ショック・加速感・ドライバビリティといった乗り心地の官能評価を実施する。また、加速性、制動性、高速安全性などの走行性能も評価し、適合の質の向上を図る。

④ 環境試験

低温、高温、低圧等の条件で、エンジン単体、車両が問題なく機能し、目標の性能が達成できるかを評価する。

以上が実験を基にした適合手順であるが、制御対象モデルを用いるモデルベース適合についても記載しておく。

上記の実験を基にした適合では、適合技術者の経験に基づく最適探索で適合を進めるのに対し、モデルを用いて適合作業の効率化を図った適合手法をモデルベース適合と呼ぶ。実験計画法、モデリング（データ処理、システム同定、実験モデル、物理モデル）、最適化アルゴリズム、制御マップ作成、自動運転・測定技術、データ・モデル・プロセス管理などの要素技術で構成され、主に上記の「①エンジン試験」、「②車両試験」に該当するところで一部の役割を担い、適合の効率化に貢献する。

次に、オートマチックトランスミッション（A/T）適合の例について紹介する。流れについては、上述のエンジン適合とそれほどの差異はない。

A/T の場合は、油圧アクチュエータの単品の特性計測から始まる。エンジンと異なり A/T のアクチュエータ部品は、この油圧アクチュエータしかない。この部品が全ての特性を決めると言っても過言ではない。まず、油圧アクチュエータ単品の電流特性、油圧特性を調整する。なるべく高速に、正しい油圧を即座に出せる duty 比、ディザ制御、フィードバック制御の定数を決定する。次に A/T 単体で温度変化できるテストベンチを使って、クラッチのストローク時間を測定する。ある程度の定数をここで決定したら、実車での適合に移行する。

エンジンと A/T は関連性の高い部品であることから、環境試験において、チームを組んで評価が行われることがしばしばある。同時評価を実施する場合、さらに第 3 者の車両適合責任者が同席し、不具合の事象に対し、どちらのパラメータを調整すべいか瞬時に判断し指示が出される。

(5) 制御システム検証

検証とは以下の二つの意味を表す。

(1) S/W、あるいはモデルが要求機能どおりに動作するか確認する

(2) S/W、あるいはモデルが真の要求を満たすものであるか確認する

英語では、Verification (1) と Validation (2) に区別している。

日本語で検証を区別する場合、(1) を検査あるいは検証、(2) を妥当性確認と呼んでいる。

(1) は入力に対して期待通りの答えが出力されることを確認する。

(2) は入力に対して出力が期待値と同一であっても、その答えを受けて動作する後の系(別の制御機能やプラントモデル)が結果的に期待通りに動作しない場合、そもそも要求が誤っていたことを示している。

S/W (あるいはモデル) の単体検査で後ろの系の結果まで知らない場合は、(1) しか検査できない。一方、同じ検査パターンで単体検査を行っても、後ろの系を接続し、後ろの系の挙動まで理解しているエンジニアが検査を行うと(2) まで含めて確認ができる。通常(1) は想定された正常なケースの検査を中心に実行し、(2) はエラーを中心として意地悪試験で発見する。モデル設計者が(1) を検査すると、思い込みによってエラーを検出できないケースがあり、近年では第3者が検証を行うケースが多くなってきている。

① 検証工程に求められるスキル

妥当性確認に求められるスキルは、技術要素スキルと直結している。特別な手法があるわけではなく、元々の真の要求を把握し、狙いとずれを勘と経験で導き出している。近年では、形式手法を用いて一致性を検証する方法が提唱されている。形式手法を用いた検証を行う場合、最終的に重要となるのはどのような項目を検証観点として設定するかであり、その抽出には決まりきった技術というものはない。

② 検証、特に機能との一致性を確認する検査について

要求されたパターンが、その通りに動作することを確認することは当たり前のことである。極論を言えば、想定される全てのユースケースを検査すればそれで終わりである。全てのユースケースをテストすることができると仮定するならば、実行するだけでそこになんら技術的なことは存在しない。しかし、実際には、「全てのユースケース」というものは、有限のパターンではない。何をパターンとするか、そして何を切り捨てるか? ここに技術的な手法が存在する。一般的なS/W開発では既に確立した手法が存在する。その最も基本的な考え方はカバレッジという方法である。ソース上の分岐条件を全て網羅することで想定されたユースケースを満たしたと仮定する方法である。

Simulink の検査で使われるカバレッジはMC/DCと呼ばれるカバレッジ方法である。MC/DCを満たすパターンを作成する場合に問題となるのは、全ての分岐を満たす検査パターンを有限時間内で調査できない可能性があるということである。特に制御設計では、状態

変数を持つ制御コントローラの設計が行われる。状態変数とは、過去の値の依存性ということであり、単純な例ではタイマーが該当する。タイマーが 10 時間経過したらという S/W があった場合、実時間でテストを行うと 10 時間が必要になる。こういった場合には、加速試験と呼ばれる手法を用いて、検査を行う。サンプリング時間を荒くして 10 時間まで早く到達させる方法、もしくはパラメータを変更して 10 時間を 10 分まで短縮する方法である。

検証のスキル項目は、JMAAB 検証標準化ガイドラインの P17 6.4.2 単体テスト仕様書記載項目を参照していただきたい。

参考

検証標準化ガイドラインより

◆ 機能一致性確認の考え方

Simulink 仕様書を基に S/W 詳細設計を進めると、内部データの量子化やプログラムの起動タイミングが原因で Simulink 仕様書とそれから生成されたコードとは完全な機能一致が得られない場合が考えられる。したがって、機能一致性に対する要求は上記のことに留意し、許容誤差内に入っていることの確認なのか、誤差要因を排除した完全一致確認なのかを明確にしておく必要がある。

機能一致性確認は生成コードだけの単体テストとレガシーコードに組込んだ状態での合体テストの両方を行う。

単体テスト報告書（あるいは合体テスト報告書）では、合否判定結果を報告するだけでなく特に不一致結果については必要に応じて要因を分析して報告する。

(6) プロセス改善

人はミスを犯すものであり、そのミスを製品に混在させないために、良いプロセスが必要である。開発・製造で問題を起こしても、発見し改良できれば製品に組込まれることはない。発見するため、また開発・製造段階で不具合を組込まないためには、良い手順を決め、決まった手順で開発・製造を行う必要がある。その決まった手順を工程もしくはプロセスと呼んでいる。

最近の考え方では、製品すなわちプロダクトの品質だけでなく、開発・製造プロセスの品質も含めて、品質というようになってきている。良い製品を作るには、良いプロセスで開発・製造した方がよい。そのため、プロセスを定義しそのプロセス通りに実践しているかの規格や、認証する機関などができており、プロセスに対する関心も大きくなっている。

プロセスの構築は一朝一夕で完成するものではない。新しい開発手法、ツールなどは日夜開発されており、それらに応じた改善が求められる。また、不具合など品質に問題が起きた

場合、製品だけでなくプロセスの見直しが必要となる。プロセスは一旦構築すれば終わりというのではなく、継続的に改善を行っていく必要がある。

5-1-4. システム開発プロセスとMBDの関係

MBDは、Simulink を用いて制御モデルやプラントモデルを設計し、検証する工程が中心となる。しかし、前後のプロセスと無関係ではない。特に Simulink を用いた設計フェーズの上流側は要求を確定するプロセスで、システムズエンジニアリングがその領域にフォーカスを当てている。近年は、システムズエンジニアリングの領域でモデルを使うようになり、モデルベースシステムズエンジニアリング（MBSE）と言われている。MBSE で使うモデルは SysML で定義された図を使っている。

E T S S - J M A A B の説明に、開発プロセスの上流側が書かれている理由は、Simulink を用いた設計の上流プロセスがあり、それはモデルベース開発を行う上では必須のプロセスであり、モデルベース開発にとってとても重要であることを示すためである。決して、SysML を採用し、システムズエンジニアリングを勉強しなさいと言っているわけではない。

しかし、システムズエンジニアリングは、開発を進める上でとても大切な知識や手法を提供してくれる。MBSE = SysML ではないし、システムズエンジニアリングは要求工程だけの設計手法ではない。例えば、自動車の開発は組織で製品開発全体をカバーしている。設計を行う部署、検査・検証を行う部署、生産を行う部署、市場の品質を調査・回収・提供する部署、最後にリサイクルと製品に関わる全てのライフサイクルまでをカバーするのがシステムズエンジニアリングである。我々はその中の設計に携わっているだけで、そこしか注目していないのが現実である。システムズエンジニアリングは、設計だけではなく、製品企画、組織構成なども含めて検討を行う。システムズエンジニアリングを勉強することは決して無駄とはならないし、MBD の設計領域とも深い関わりがある。重要なことはこれらの工程を意識し、設計フェーズにおいても、要求そのものが正しいか常に考え、要求の真の意図を読み取ることである。間違っているならば上流側に戻すことを恐れてはいけない。モデルベース開発は素早く上流側に戻すことができることもメリットの一つである。要求に一致するよう下の工程だけを素早く何回も回してもうまみはないのである。そういった上流側に立ち返り、お客様のための良い製品を開発することを意識してもらうために Simulink の設計工程より上流側も記載している。

II. ETSS-JMAABの概要

本章では、本ドキュメントの位置づけや構成について概要を説明する。

ETSS-JMAABに関する、必要性や構成などの全体的な基本事項の説明を行う。

1. 組込みスキル標準（ETSS）との関係

組込みS/Wエンジニアの育成を目的に情報処理推進機構で作成された『組込みスキル標準（Embedded Technology Skill Standards、以下「ETSS」と略称）』を基にして、そこに書かれた手法に注目し、自動車業界向けに使うにはどのようにすれば良いのかその手順について記載したのが2008年公開のETSS-JMAABである。

2. ETSS-JMAABとは

2-1. ETSS-JMAABの概要

MBDによる自動車制御系開発力の強化に向けた『ETSS-JMAAB』は、JMAAB MBDエンジニア育成WGによって2008年7月にV1.1が公開された。ETSS-JMAABは、経済産業省によって設置された、組込みソフトウェアエンジニア開発力強化推進委員会にて、組込みソフトウェアエンジニアの育成を目的に作成されたETSSをもとに作成されている。ETSS-JMAABは「スキル基準」、「キャリア基準」、「教育研修基準」によって構成されている。自動車用制御装置のMBDを進めていくエンジニアの育成やエンジニアの有効活用を目的に、モデルベースの制御システム開発におけるモデリングやツールに関する知識、設計プロセスにおける活用手法などを整理し、対象となる職種を定義した上で、MBDエンジニア育成のための指南書となるものである。

2-2. ETSS-JMAABの構成

ETSS-JMAABは、自動車制御システム開発に関するスキルを中心とした、人材の育成や活用などに関する要素を体系的に以下のように3つの部分で整理分類されている。

- ◆ スキル基準 開発対象の制御システム開発スキルを体系的に定義する。
- ◆ キャリア基準 開発対象の制御システム開発に関わる職種/専門分野を定義する。
- ◆ 教育研修基準 キャリア基準に到達させるための仕組みを定義する。ETSS-JMAABでは「教育研修基準補足」として表している。

スキル基準は、自動車制御システム開発に必要なスキルを、大きく技術要素スキル、開発スキル、管理スキルの3つに分類する。個々のスキルについて言及はしないが、スキル定義の手順などを記述している。

キャリア基準は、自動車制御システム開発に関わる職種/専門分野を定義して、それぞれの職種に必要なスキルを明確にするものであり、各職種のエンジニアが個別にどのスキルを向上させるべきかを考える際の指針となる。また、制御系開発組織の構成員のスキルを全体的に評価し、組織全体としてどのスキルが不足しているかあるいは重複しているかなどを客観的に判断するための指針としても利用できる。

教育研修基準は、ETSSでは組み込みS/W開発の人材育成を実現するための教育や訓練に関する構造や仕組みを定義する。教育プログラムで履修する内容を、スキル基準で整理されたスキル項目を用いることで研修内容や対象レベルの枠組みを提示するものである。

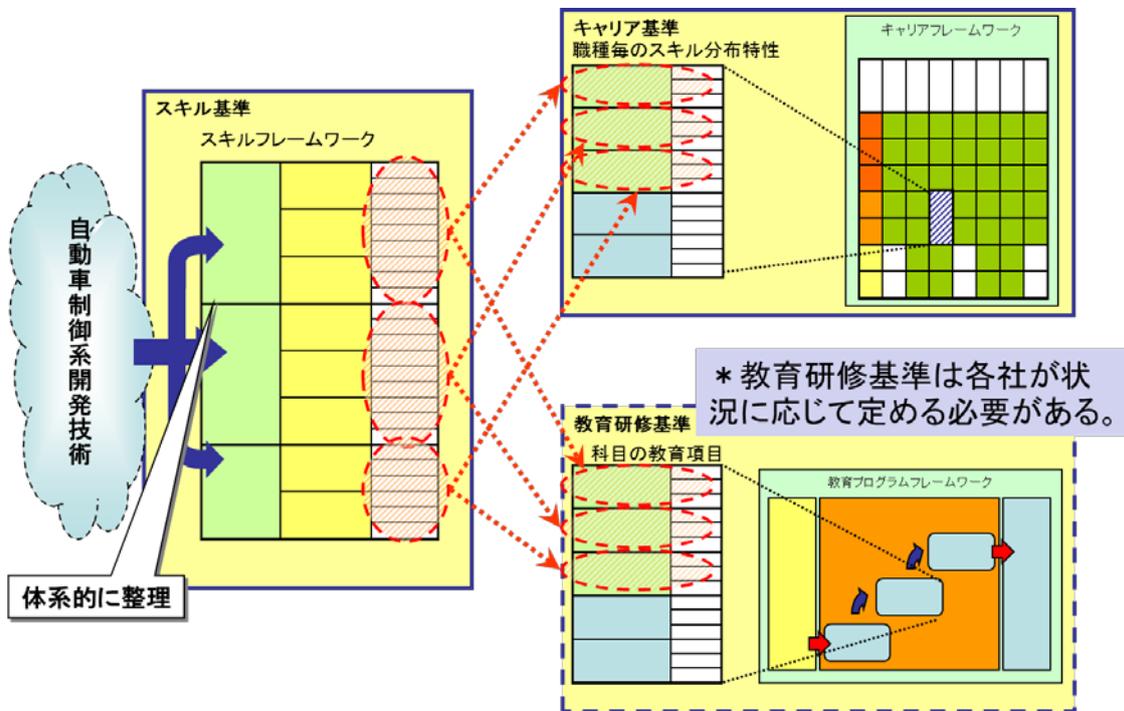


図 6. ETSS-JMAABの構成

2-3. スキルレベルとキャリアレベルの考え方

ETSS-JMAABでは、スキル基準とキャリア基準に、それぞれスキルレベルとキャリアレベルと2つの評価指標が存在する。ここでは、この2つの特性について説明を行う。

2-3-1. スキルレベル評価の考え方

スキル基準におけるスキルレベルとは、スキルカテゴリごとに階層的に分類された技術項目ごとに、評価対象の開発者個人もしくは組織が発揮するスキルのポテンシャル（期待値）の度合いを数値で表現したものである。本来、個人や組織が持つスキルは多種多様であり、技術項目ごとにスキルポテンシャルは異なるものである。したがって、開発者個人や組織の持つスキルを評価するためには、「図 7. 開発者のポテンシャルを複数のスキルレベルで可視化する」のように、複数の技術項目ごとにスキルレベルの測定を行い、その分布状況を分析する必要がある。

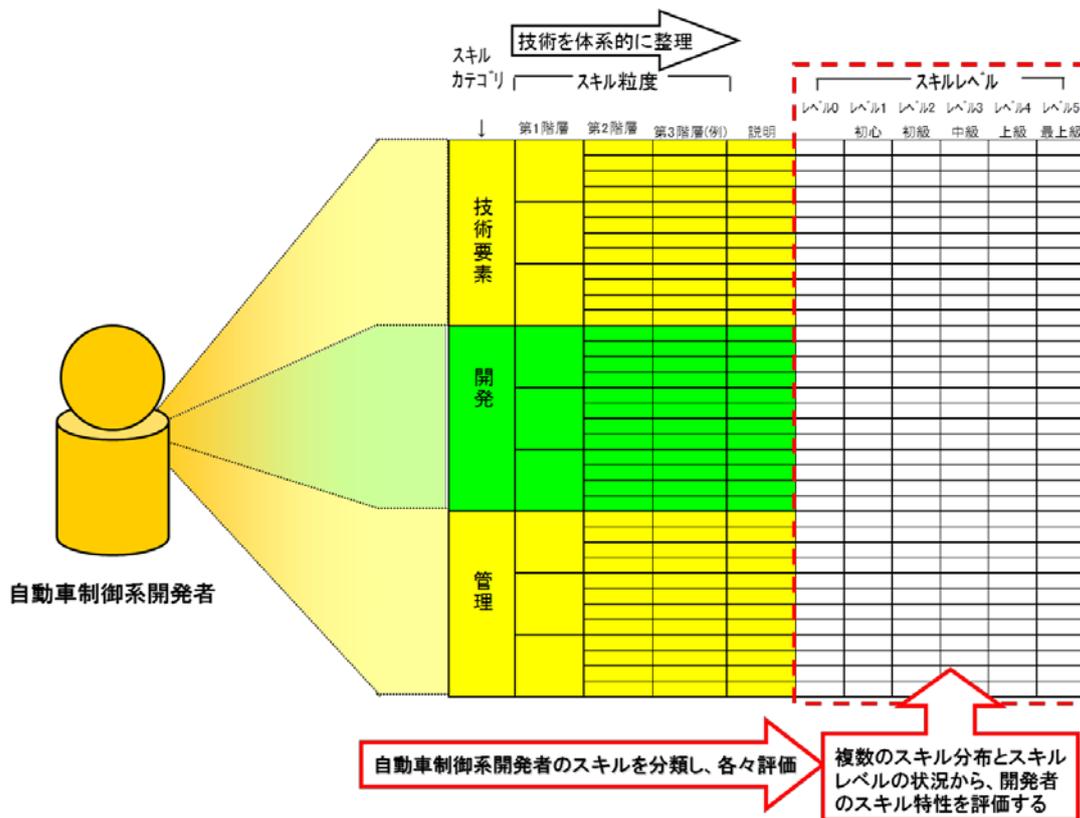


図 7. 開発者のポテンシャルを複数のスキルレベルで可視化する

2-3-2. キャリアレベル評価の考え方

キャリア基準におけるキャリアレベルとは、組織やプロジェクトの中で職種が果たすべき責任に対するビジネスやプロフェッショナルの貢献の度合いを1つの評価軸で表している。つまり職種とそのキャリアレベルは技術者個人が、組織やプロジェクトの中で役割や責任を、どの程度果たすべきか、あるいは達成できたのかを、1つの指標（キャリアレベル）で評価するものである。キャリアレベルは、先行するETSSのスキル標準を踏まえた形で、3段階（エントリー、ミドル、ハイ）のレベルとなっている。（ETSSでは7段階で評価する方法もある。）

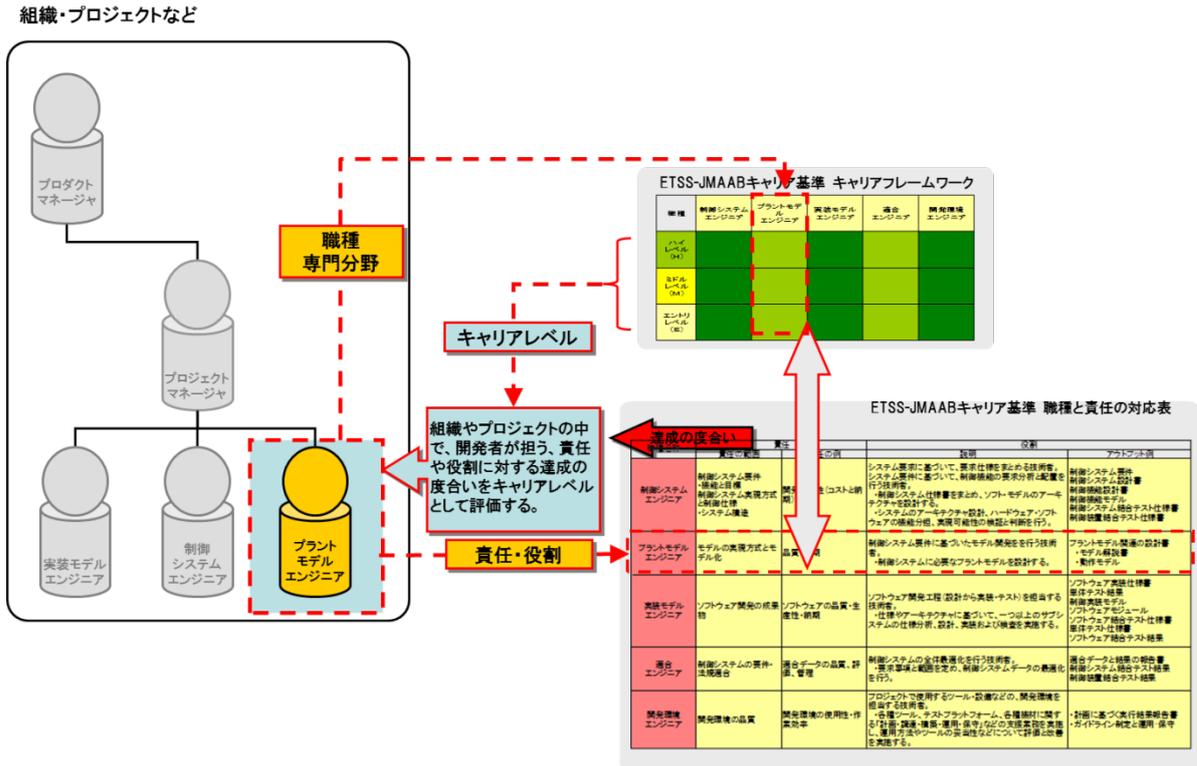


図 8. 職種や専門分野が担う責任や役割の達成度合いをキャリアレベルとして可視化

ある人材に対して組織内の役割を割り当てるなどの人材活用を考える場合、MBDに関する職種と3段階のキャリアレベルごとにスキルの分布特性を作成する必要がある。このスキル分布特性と個人の、スキル分布特性を比較することで必要となるスキル項目を客観的に評価できる。また人材育成のための不足技術項目の可視化も可能である。

スキルの分布特性により、実務の必要性に応じたスキルレベルの範囲を示すことができる。つまり、キャリア基準を運用する企業において、方針・運用で該当する個所のスキルレベルの点数をこの範囲で定義することが可能となる。

2-4. ETSS - JMAABに期待される効果

ETSS - JMAABは次のような状況での利活用が期待できる。

2-4-1. 個人における利活用

■ 技術者としての強みと弱みを認識

スキル基準を用いて技術者個人のMBDに関するスキル測定とスキルレベルの分布を可視化することができる。スキルレベルの分布から、技術者の強みや弱みを客観的かつ定量的に認識することが可能となる。

■ 人材育成

スキル基準を用いることで育成ビジョンを示し、育成対象を明示することが可能となる。また、育成状況の確認においては、育成前と育成後のスキルアップが可視化され育成に関するモチベーションを確保することにも効果が期待できる。

■ 具体的なキャリアパスの確認

スキル測定された技術者自身のスキルレベル分布状況と、キャリア基準で定義された職種/専門分野のキャリアレベルの関連スキルとを付き合わせることで、現状の職種/専門分野におけるキャリアレベルの妥当性を確認できる。

また、目標とする職種のキャリアレベルまでのキャリアパスをどのような経路で、どのようにレベルアップしていくのかを具体的にイメージすることもできる。

2-4-2. 組織における利活用

■ スキルに関する開発リスクの分析

開発対象製品に必要なスキルレベルの分布と、開発チームのスキル分布を比較することでスキル不足による開発プロジェクトのリスク分析を行うことができる。スキルの不足部分を定量的に可視化することで、追加要員に必要なスキルのリストアップを行うなどの的確なリスクヘッジプラン策定のための指針にもなりうる。

キャリア基準、教育研修基準育成基準に基づき、開発チームメンバーのキャリアアップ及びスキル向上のため教育・育成を行うための指針とできる。

■ 人材育成での活用

人材育成においては、スキル基準を用いることで組織全体としての育成ビジョンを示し、育成対象を明示することが可能となる。

■ プロジェクトでの活用

スキル基準は開発者個人や組織（企業やチームなど）のスキルを可視化するものであり可視化された情報は、以下のようなシーンで活用される。

- プロジェクト計画 : 人員配置などの具体化、明確化
- 採用・調達 : 上記「計画」に従った人材の採用・調達
- 業務遂行 : 上記「計画」に従い上記「採用・調達」された人材（スキル）による業務遂行
- 評価(*) : 業務の遂行状況や結果を評価し、組織や個人の行動にフィードバック

2-4-3. 運用における留意点

スキル基準を人材の評価や処遇に適用することについては、「スキルの可視化による企業競争力の強化と人材育成」という本来主眼とする趣旨とは異なるものである。運用方法などによっては、本質を忘れた手順習得に特化した、考えるスキルの低下を招く恐れがある。したがって、ETSS－JMAABの運用にあたっては、下記などが起きかねないように充分留意が必要である。

- ◆ 技術者のモチベーションを減退させ、結果としてマイナスインパクトを与えかねない。
- ◆ 表面的なスキルをあげることにとらわれ、モノづくりの本質を見失い、マニュアル人間になる可能性がある。

技術者のモチベーションを上げるためには、目指す目標を見失わないようにする必要がある。そのためには、明確なキャリアパスを描き、必要なスキルを正しく定義しなければならない。

マニュアル人間にならないようにするには、まず作業と知的創作は異なることを忘れないようにすべきだ。エンジニアの本質は知的創作である。「考える」・「工夫する」などエンジニアの本質を忘れてない教育を心がけ、教育コンテンツを作る必要がある。

III. スキル基準

本章では、スキルの定義等について説明している。

1. ETSS－JMAABスキル基準の概要

ETSS－JMAABスキル基準は、MBDによる自動車制御システム開発に必要なスキルを明確化・体系化したものであり、人材育成・活用に有用な指標（共通基準）を提供しようとするものである。

MBDによる自動車制御システム開発に必要なスキルは多岐にわたるが、スキル基準ではビジネスやパーソナルなどのスキルは定義していない。また、会社、職種、ドメイン等に応じて必要なスキルは異なるため、必要なスキルを抽出して利用者がスキル基準を作成することが望ましいため、本章のスキルは例としての掲載である。重要なことはスキルを分解し体系づけ、自社のスキルを定義づけることであり、本章はその考え方を示している。

2. ETSS－JMAABスキルフレームワーク

自動車業界では高い安全性・信頼性・リアルタイム性が要求される一方で、制御の多様化やハードウェア性能の向上を背景にシステムの大規模化・複雑化が進んでおり、MBDはこれらに対応する効率の良い開発手法として導入が進んでいる。さらに技術の進歩も早く、製品の陳腐化も早くなった。このため、MBDに関わるエンジニアには、製品に対して新しい技術をタイムリに取り込むスキルを求められている。

MBDによる自動車制御系開発に必要なスキルを体系的に整理し、明確にすることで、人材育成と活用を推進して高品質な製品を短期間に開発することが可能となる。

2-1. スキルの定義

スキルとは熟練や技能と表現されることが多い。スキル基準では、スキルとは作業の遂行能力を指し、「～ができること」を表現するものであり、知識を有するだけではスキルとは扱わない。知識はスキルを発揮するために必要な構成要素で、特に自動車の製品知識と工学基礎知識はMBDエンジニアに必要な知識である。スキルと知識の関係は「図 9. スキルと知識の関係」のように示される。

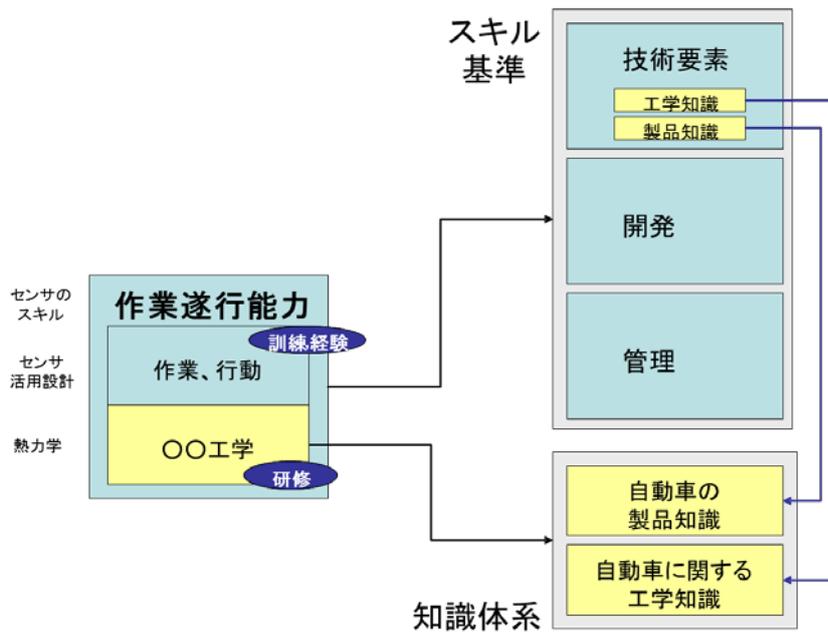


図 9. スキルと知識の関係

例えば、自動車に関する工学基礎知識、製品知識は「表 1. 自動車に関する知識カテゴリ」のように分類することもできる。

表 1. 自動車に関する知識カテゴリ

| 第1階層 | 第2階層 | 第3階層 (例) | 説明 |
|------|--------|--------------|--------------------------|
| 1 | 工学基礎知識 | 1 熱力学 | ・一般工学基礎 |
| | | 2 流体力学 | ・一般工学基礎 |
| | | 3 運動力学 | ・一般工学基礎 |
| | | 4 機械 | ・一般工学基礎 |
| | | 5 電磁気学 | ・一般工学基礎 |
| | | 6 電子工学 | ・一般工学基礎 |
| | | 7 ソフトウェア工学 | ・一般工学基礎 |
| | | 8 制御工学 | ・一般工学基礎 |
| | | 9 数学・工業数学 | ・一般工学基礎 |
| | | 10 統計学 | ・一般工学基礎 |
| | | 11 人間工学 | ・一般工学基礎 |
| | | 12 自動車工学 | ・一般工学基礎 |
| 2 | 製品知識 | 1 車体系製品 | サンルーフなど車体系製品知識 |
| | | 2 パワートレイン系製品 | エンジン、ATなど駆動系製品知識 |
| | | 3 ITS系製品 | ナビゲーションなどITS系製品知識 |
| | | 4 シャーシ系製品 | ABS、サスペンションなど走行系製品知識 |
| | | 5 その他 | モータ/発電機、電池、FC、HVなど構成要素知識 |

2-2. ETSS - JMAAB スキルフレームワークの構造

ETSS - JMAAB スキルフレームワークは、「図 10. ETSS - JMAAB スキルフレームワーク」のような構造を持ち、MBD 技術を整理することを目的としている。

| スキル カテゴリ | スキル粒度 | | | | スキルレベル | | | | | |
|-------------|-------|------|---------|----|------------|------------|------------|------------|-------------|------|
| | 第1階層 | 第2階層 | 第3階層(例) | 説明 | レベル0 初心 | レベル1 初級 | レベル2 中級 | レベル3 上級 | レベル4 最上級 | レベル5 |
| 技術要素 | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| 開発 | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| 管理 | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |

図 10. ETSS-JMAABスキルフレームワーク

■ スキルカテゴリ

自動車制御システムMBDに必要な技術を「技術要素スキル」「開発スキル」「管理スキル」の3つに区分する。

■ スキルレベル

階層的に整理された技術に対する作業遂行能力の期待値を6段階で表したものである。

2-3. ETSS-JMAABスキルカテゴリの説明

ETSS-JMAABスキルフレームワークのスキルカテゴリは、自動車制御システムMBDに関する技術を体系的に分類・整理するための起点となる。

■ スキルの整理

スキルを「技術要素スキル」「開発スキル」「管理スキル」のスキルカテゴリで整理する。

技術要素スキル : 対象とするシステムの機能を実現する技術項目

開発スキル : 対象とするシステムの開発時に使用する技能等

管理スキル : 対象とするシステムの開発を円滑かつ的確に進行させるために使用する手法を使いこなす技能等

■ スキルの詳細化、具体化

スキルカテゴリを起点として階層的に第1階層から第n階層へと自動車制御システムMBDの技術を詳細・具体化する。目的によって分解される階層、粒度は異なる。教育を目的とする場合は、教育可能な技術や知識まで分解する必要がある、人材評価や人材登用などに使用する場合は、個々の技能が解るように分解する必要がある。

2-4. ETSS-JMAABスキルレベルの説明

2-4-1. ETSS-JMAABスキルレベルの定義

ETSS-JMAABスキルフレームワークのスキルレベルは、各スキルカテゴリ共通の定義を持つ。

ETSS-JMAABでは、スキル項目ごとに作業遂行能力の期待値（ポテンシャル）をスキルレベルで表現する。技術要素スキルレベルは6段階で表す。1（初心）～4（上級）は、確立された技術に関する作業遂行能力の度合いを定義し、それに加えて技術革新（イノベーション）を推進できる能力を評価するために、最上級のスキルレベル5、不要スキルとしてのレベル0を定義している。スキルレベルは常に6段階全てを使う必要はない。例えば、開発スキルレベルは4段階を用いて、スキル項目の内容によって定義も若干変化する。

■ 技術要素スキルの例

- ◆レベル5：最上級 新たな技術を開発できる
- ◆レベル4：上級 作業を分析し改善・改良できる
- ◆レベル3：中級 自立的に作業を遂行できる
- ◆レベル2：初級 支援の元に作業を遂行できる
- ◆レベル1：初心 内容を知っている。（スキルの出発点を示す）
- ◆レベル0： 内容を知らない。（キャリア基準のスキルレベルマップにおいては不要スキルを示し、マップ上は空欄として何も記入しない）

■ 開発スキル（図表記のツール）例

- ◆レベル3：上級 分析・解析・活用できる
- ◆レベル2：中級 使える
- ◆レベル1：初級 読める
- ◆レベル0： 内容を知らない。

2-4-2. ETSS-JMAABスキルレベルの評価要件

「スキル（業務遂行能力）がある」ということの要件を明確に定義するのが、スキル評価要件である。個々のスキルについて、具体的な評価要件を提供するアプローチもあるが、ETSS-JMAABのスキルフレームワークでは共通的な評価要件を提供する。

■ 技術要素スキル

作れる：「開発スキルを使って、〇〇を実現することができる」

〇〇：技術要素名称

■ 開発スキル

使える：「□□の××を使うことができる」

技術要素のみでスキルは定義できるが、複数の技術要素で共通の手法などを取り出したものが開発スキルと言える。

□□：開発スキルの第一階層。第一階層の開発プロセス名称。

××：開発スキルの第二階層。第二階層以下の開発技能、開発手法、開発ツール名称。

■ 管理スキル

使える：「××を使って□□ができる」

* ETSS-JMAABでは言及しない。

□□：管理スキルの第二階層。第二階層の管理プロセス名称。

××：管理スキルの第三階層。第三階層の管理手法、管理ツール名称。

評価要件で使用している「～ができる」には動作と知識に関する2つの視点が必要となる。「～ができる」ということは、実際に動作としての作業が行えるということの意味する。作業を行う際には、「正確性」や「効率性」などが基本的に求められ、さらに適切な「状況判断」といった応用力も求められる。

また、このような動作をするための前提として、作業に使用する手法やツールに関する知識が必要となる。したがって、手法やツールを使う対象物や環境、手順などに関する知識も必要である。

これらの動作や知識をチェックすることで「～ができる」ということを判断することができるようになる。

スキル評価は評価方法と評価体制、特に技術項目について高い知見を有しているかどうかといった評価者の適格性が重要である。開発業務の実施状況からスキルのレベルをどのように評価するのかについて十分に検討し明示する必要がある。

2-4-3. ETSSとの表現の違い

ETSSでは、技術要素スキル、開発技術スキル、管理技術スキルの3つのカテゴリに分ける。ETSS-JMAABでは、開発、管理のスキルは、技能に近い内容や手法を示すので、技術ではないと位置づけた。したがってカテゴリ名称からも技術を削除している。技術に関するスキルは全て技術要素スキルに分類する。

3. ETSS - JMAAB スキル基準

各企業において実務ベースでの活用を目指すには、スキルを計測できるまで具体化することが必要である。各社の技術要素スキルマップへ統合する、あるいは開発スキルマップだけを活用するなど、定義の参考としていただきたい。スキル基準で定義する技術の範囲は、MBDによる自動車制御システム開発で共通的に利用されるものを想定している。各企業や応用ドメインで利用される固有の技術に関しては扱っていない。

このような固有の技術要素スキルに関しては、各企業や応用ドメインの団体・グループにて標準化を行い、固有スキルの扱いを検討して欲しい。広い範囲で利用可能な開発スキルは公開し、企業が必要とするスキルを持った人材の獲得を実現することや、MBDによる自動車制御システム開発に共通のものとしてスキル基準に反映することも可能である。

3-1. 技術要素スキルカテゴリ

技術要素スキルは、対象とするシステムの機能を実現する技術項目であり、各社が定義すべきである。各社がどのような単位で運用するかも検討して分類する必要がある。ここには具体的なドメインに対する技術要素スキル例を記載した。

表 2. 技術要素スキルカテゴリ(CVT制御開発の例)

| | | | 部 | 制御開発部 | | | | S/W開発部 | | | | |
|------|--------|------|--------|-------------|----|-------|----|------------|----|-------|----|--|
| | | | | 制御システムエンジニア | | | | 実装モデルエンジニア | | | | |
| | | | チーム | 機能1制御 | | 機能2制御 | | 機能1実装 | | 機能2実装 | | |
| | | | 個人 | 山田 | 石田 | 鈴木 | 加藤 | 片山 | 児玉 | 山本 | 小島 | |
| 技術要素 | 第1階層 | 第2階層 | 第3階層 | | | | | | | | | |
| | 制御要求 | 変速制御 | 通常変速 | | | | | | | | | |
| | | | 急速ダウン | | | | | | | | | |
| | | 油圧保持 | 前後クラッチ | | | | | | | | | |
| | | | 安全油圧 | | | | | | | | | |
| | その他 | 故障 | | | | | | | | | | |
| | 制御 S/W | 変速制御 | 通常変速 | | | | | | | | | |
| | | | 急速ダウン | | | | | | | | | |
| | | 油圧保持 | 前後クラッチ | | | | | | | | | |
| | | | 安全油圧 | | | | | | | | | |
| | | その他 | 故障 | | | | | | | | | |
| | | | 通信 | | | | | | | | | |

■ 表の解説

技術要素スキルは、対象とするシステムの機能を実現する技術項目になり、システムの機能を実現する技術項目は、製品や、図面等が該当する。技術要素を、部品レベルに分けていくと、階層の切れ目は、部署、グループの切れ目になる場合が多いのではないと思われる。

ここでの例も、第1階層がプロセスの切れ目になっているように見える。例では、制御設計部署と、S/W設計部署という部門で分かれている。制御要求の部署は、制御要求（仕様書、図面）を設計する。制御S/Wの部署は、実装モデルの設計やS/Wの設計を担当している。

3-2. 開発スキルカテゴリ

開発スキルは、対象とするシステムの開発時に使用する技能、手法等のことである。各社は目的に応じて細かいプロセスの切れ目を明確にし、そこでのアウトプットを明示的に示す必要がある。（職種よりも小さい単位のプロセスをイメージした方が良い。）

① プロセスごとに求められる成果物を設計するために必要とする技能を抽出し、それを開発スキルとする。

開発スキルは特定のプロセスだけで使われるスキルと、各プロセスで横断的に使われるスキルがあると考えられる。また技術要素にも特定のスキルと、どのような技術要素でも横断的に使われるスキルもある。

例えば見逃しがちなスキルとして Microsoft Word を使いこなすスキルがある。実は何らかのドキュメントを作る場合、必ず必要になるスキルであるが、このスキル、Word のバージョンにも依存している。Microsoft Office2003 から 2010 にバージョンアップすると、その日、仕事ができなかったという話は良く聞くことである。単純であるが、このような基本部分も開発スキルの一部であり、細かく分類すれば、Word2003 と Word2010 は少し異なるスキルとして定義することもできる。これらは、各社が何を意図してスキルマップを作成するかによって、スキルの項目、詳細度は異なることを意味している。

② 開発プロセスを明確にし、成果物を定義し、その成果物を設計するために必要とされる知識、技能を分離し、スキル項目、スキルレベルを定義する。

スキル項目、スキルレベルはその定義の目的によって詳細が異なる。具体的には、単に人材を評価したいだけなのか、個人の教育カリキュラムの計画をしたいのか、などと言った目的があげられる。E T S S を用いた人材の可視化は、目的があってその目的に応じた詳細度が必要である。場合によっては目的に応じて複数の詳細度で管理しなければならない。

具体的な抽出の例では、シナリオを分析して戦略を作るとか、要求を抽出してやるべきことを整理することはスキルだと考えられるが、残念ながら、頭の中だけで戦略を作って、行動し

ても許されないのである。熟練者の「瞬間のひらめき」と言われる領域であるが、これらも「他の社員がどうやってそれらの手法を使えるようにするか」までを分解、落とし込む必要がある。つまり、一つ一つのスキルを

- ・ どうやって評価するか？
- ・ どうやって教育するか？

に繋がる手法まで落とし込む。

例えば、一般的には戦略や分析は、成果物を作る過程で使われるスキルになる。設計の成果物そのものではないにしろ、それらは途中で表や図を使ってまとめられ、まとめた表や図を使って分析している。そのようなことをしなくても、「ひらめく」と言われると、ETSSの仕組みの全否定になる。教育対象者をそれらの領域に到達させるためには、先人たちが残した何らかの手法があるはずであり、そういった分析や戦略は、ある特定の「表」や「図」が書ける、設計できることが、開発スキルとして具体化でき、手法を教えることで、育成できるはずである。

具体的には、タイミングチャート、シーケンス図、構成図、FTA、要因解析図、QC7つ道具や、スタックアップ、重回帰分析とか、多変量解析、そういった手法がたくさんある。どこに何をを使うのかは、目的やドメインによって異なるので、より具体化した方が良い。

そういった具体的な手法に落とせるように、開発スキルを定義する必要がある。

きちんと分解、定義ができれば、開発スキルは講座などの提供によって、知識を提供し、簡単な演習によって短期的に身に付けることができるスキルである。しかし、技術要素スキルは単純ではない。たくさんの開発スキルの習得と実践による育成が必要となり、一般的に技術要素スキルはOJTによって長期間かけないと育成できないと言われている。

この育成の違いからも、開発スキルと技術要素スキルの違いとしてまとめ方を検討してほしい。

そして、突き詰めると、技術要素スキルと、職種が一致する場合がある。それはドメインが非常に狭い場合に起こり得るまとめ方である。通常は一製品開発にしても、将来はさらに複数の因子に分解され、チームごとに分かれるものである。職種とはそれらのチーム横断で定義できる抽象的な存在で、チーム移動などを経て、職種のレベルが上がるものである。

開発スキル、技術要素スキル、キャリア基準と、それぞれ表現している違いを明確にし、まとめて欲しい。

3-3. 管理スキルカテゴリ

このカテゴリは、必要であれば、ETSSを参照してください。MBD特有の技術とは考えられないため、ETSS - JMAABでは言及しない。

IV. キャリア基準

本章では、MBDエンジニアに関する職種名称や職掌定義をした『キャリア基準』について記載している。

1. ETSS－JMAABキャリア基準とは

1-1. ETSS－JMAABキャリア基準の概要

ETSS－JMAABキャリア基準とは、MBDに従事する技術者（MBDエンジニア）の主な職種、その内容を明示したものである。キャリア基準は、職種についての自動車業界共通の名称として使われることを意図している。各社の利用目的に応じて、キャリア基準・職種の細分化が必要となる。

1-2. ETSS－JMAABキャリア基準の必要性

現状、自動車制御システム開発に従事する技術者の肩書きは、職制上の肩書きとなっており、技術的な役割に対応していない。技術者の求人においても、「制御システム」「組込みシステム」など広範囲なドメイン指定の他は、「設計者」、「プログラマ」、「システムエンジニア」、「S/W開発」などの大きくりの職種が指定されているのみであることが通常である。

このような現実には、制御系技術者の技術的役割が未分化で、求められる専門性として何かあるかが明確になっていないことを物語っている。その結果、求人においては、経験年数や固有名詞として例示された技術の開発経験が、応募資格や望ましい経験・能力となることになる。現状では、技術者は、基本的に、自らが参加した開発プロジェクトで経験した技術のみが証明できる能力であり、技術者が自立的に専門性を深める指針が乏しいことになる。

キャリア基準は、MBDを用いた自動車制御システム開発において必要な技術者の主な職種を明示し、各職種の概要、求められるスキルを合わせて示したものである。

1-3. ETSS－JMAABキャリア基準の期待される効果

1-3-1. 個人にとってのメリット

キャリア基準が示されることにより、個人は、MBDエンジニアとしての将来の可能性を俯瞰することができ、自らの適性や環境に即して、技術要素スキル、開発スキル、管理スキルなどを含む総合的な能力開発を図ることができ、自らの目標が明確になる。

1-3-2. 企業にとってのメリット

職種（役割）が明らかになることにより、特定の開発プロジェクトにおいて必要な技術的役割や技術者の数を、より精密に見積もることができるようになる。また、必要に応じて技術者の採用時や、一時的に外部技術者を調達する際に、求める能力をこれまで以上に正確に表現することが可能になる。

より長期的には、事業戦略に沿って社内で増強すべき職種を明確化すれば、そのために必要な人材計画（採用、人材育成、外部調達など）を合理化できることになろう。

2. キャリア・フレームワーク

キャリア基準を記述するための枠組みをキャリア・フレームワークと呼ぶ。キャリア・フレームワークでは以下を提示している。

- ◆ 職種の区分
- ◆ キャリアレベルの定義
- ◆ 職種とスキルとの対応付け
- ◆ 職種の活動領域

2-1. 職種の区分

M B Dに関連する職種を「職種」として区分し、それぞれの職種の概要と役割・責任を記述し、キャリアレベルを定義する。

M B Dエンジニアの職種は以下の5種類を設定する。

- ◆ 制御システムエンジニア
- ◆ プラントモデルエンジニア
- ◆ 実装モデルエンジニア
- ◆ 適合エンジニア
- ◆ 開発環境エンジニア

2-2. キャリアレベルの定義

キャリアレベルは、当該職種において人材がプロフェッショナルとしての価値創出に応じたスキルの度合いを3段階のレベルで表し、エントリレベル、ミドルレベル、ハイレベルと呼び、職種に共通して、下記を意味する。

M B Dキャリア基準では、人材育成やプロジェクト要員の調達を想定しているため、3段階のレベルと定義した。用途により拡張が必要な場合は、企業内の責任で利用目的に応じて拡張する必要がある。

◆ ハイレベル

社内外の他の職種との調整を実施でき、当該職種に係るテクノロジーやメソドロジ、ビジネスをリードするレベル。さらに、社内の人材投資戦略に関して策定・実行に大きく貢献することが求められる。

◆ ミドルレベル

社内外の他の職種の内容を把握し、自らのスキルを駆使することによって、業務上の課題の発見・解決を行うことができるレベル。また、下位レベルの育成に積極的に貢献することが求められる。

◆ エントリレベル

当該職種の上位レベル指導の下で、業務上における課題の整理と解決を行うことができるレベル。

2-3. 職種とスキルとの対応付け

職種とスキルとの対応付けは、各社のニーズに応じて設定する。ここで、対応付けされるスキル項目は、各社の技術要素スキル、開発スキル、管理スキルで示されたスキルレベルを参照する。

2-4. 職種の活動領域

職種ごとに実業務における活動領域を提示する。

前述の内容を図示すると「表 3. キャリアレベルの定義」の通りである。また、各レベルにおける、プロフェッショナルとして要求される経済性と責任性に対する達成指標を参考的に図表化すると「表 4. キャリアレベルとプロフェッショナル評価の達成指標」の通りとなる。

表 3. キャリアレベルの定義

| | | エントリレベル | ミドルレベル | ハイレベル |
|---------------------|--------------|----------|----------------------------|----------|
| 要求作業 (役割) の達成 | 価値創造 への貢献 | | | 社内をリードする |
| | | | 経験を知識化し、業務の改善や後進育成の面で応用できる | |
| | | | 独力ですべてできる | |
| | | 指導の下でできる | | |

表 4. キャリアレベルとプロフェッショナル評価の達成指標

| | エントリーレベル (E) | ミドルレベル (M) | ハイレベル (H) | 貢献度 高 ↑ ↓ 低 |
|--|--------------|------------|-----------|-------------------------|
| ◎プロフェッショナルとして要求される経済性と責任性 ・新規性 ・ミッションクリティカル性 ・大規模/複雑性 ・国際性 ・専門技術の向上、普及 ・倫理性 ・人材マネジメント支援 ・教育訓練支援 ・メンタリング ・・・・ | | | | |

3. キャリア基準

本節では、職種とキャリアレベル及びその対応付けの詳細を説明する。

3-1. 職種とキャリアレベル

M B D エンジニアの職種は前述した通り 5 種類制定しており、各職種のキャリアレベルは、「図 11. 職種とキャリアレベル」の通り、エントリーレベル、ミドルレベル、ハイレベルで分ける。

| 職種 | 職種 | 制御システムエンジニア | プラントモデルエンジニア | 実装モデルエンジニア | 適合エンジニア | 開発環境エンジニア |
|---------|--------------|-------------|--------------|------------|---------|-----------|
| キャリアレベル | ハイレベル (H) | | | | | |
| | ミドルレベル (M) | | | | | |
| | エントリーレベル (E) | | | | | |

図 11. 職種とキャリアレベル

3-2. 職種のスキルレベルとの対応付け

職種のキャリアレベルごとに、スキルレベルマップを各社において作成する必要がある。それぞれの職種には、主として担う分野（技術領域）がある。この技術領域で必要とされる基本スキルレベルの点数を決める。

職種としては別々であるが、一人のエンジニアが複数の職種を兼ねるという場合も現実にはある。個人としての能力ではなく、その職種のそのキャリアレベルに必要なスキルを明確にすることが必要である。

3-3. 職種と責任及び役割の対応

表 5. 職種と責任及び役割の対応表

| 職種名称 | 責任 | | 役割 | |
|--------------|--|------------------|--|---|
| | 責任の範囲 | 責任の例 | 説明 | アウトプット例 |
| 制御システムエンジニア | 制御システム要件 ・機能と目標 制御システム実現方式と制御仕様 ・システム構造 | 開発効率性(コストと納期)・品質 | システム要求に基づいて、要求仕様をまとめる技術者。システム要件に基づいて、制御機能の要求分析と配置を行う技術者。 ・制御システム仕様書をまとめ、ソフト・モデルのアーキテクチャを設計する。 ・システムのアーキテクチャ設計、ハードウェア・ソフトウェアの機能分担、実現可能性の検証と判断を行う。 | 制御システム要件 制御システム設計書 制御機能設計書 制御機能モデル 制御システム結合テスト仕様書 制御装置結合テスト仕様書 |
| プラントモデルエンジニア | モデルの実現方式とモデル化 | 品質・納期 | 制御システム要件に基づいたモデル開発を行う技術者。 ・制御システムに必要なプラントモデルを設計する。 | プラントモデル関連の設計書 ・モデル解説書 ・動作モデル |
| 実装モデルエンジニア | ソフトウェア開発の成果物 | ソフトウェアの品質・生産性・納期 | ソフトウェア開発工程(設計から実装・テスト)を担当する技術者。 ・仕様やアーキテクチャに基づいて、一つ以上のサブシステムの仕様分析、設計、実装および検査を実施する。 | ソフトウェア実装仕様書 単体テスト結果 制御実装モデル ソフトウェアモジュール ソフトウェア結合テスト仕様書 単体テスト仕様書 ソフトウェア結合テスト結果 |
| 適合エンジニア | 制御システムの要件・法規適合 | 適合データの品質、評価、管理 | 制御システムの全体最適化を行う技術者。 ・要求事項と範囲を定め、制御システムデータの最適化を行う。 | 適合データと結果の報告書 制御システム結合テスト結果 制御装置結合テスト結果 |
| 開発環境エンジニア | 開発環境の品質 | 開発環境の使用性・作業効率 | プロジェクトで使用するツール・設備などの、開発環境を担当する技術者。 ・各種ツール、テストプラットフォーム、各種機材に関する「計画・調達・構築・運用・保守」などの支援業務を実施し、運用方法やツールの妥当性などについて評価と改善を実施する。 | ・計画に基づく実行結果報告書 ・ガイドライン制定と運用・保守 |

※補足 (表中の用語説明)

制御システムエンジニア

- ◆ システム要求：機能の性能など曖昧さが残る表現が含まれているもの
- ◆ システム要件：システム要求から具体的な値や設計値として明確化されているもの

3-4. 職種

MBDエンジニアの5種類の職種について、以下に説明する。

3-4-1. 制御システムエンジニア

【職種の説明】

製品機能を実現するための制御機能モデルを設計する技術者

【職種の役割】

- 製品機能に対応した制御システム特性の定義を行い、制御機能モデルを設計する
- 制御システムの開発効率（コストと納期）と品質を確保する
- 制御システムに必要となるセンサやアクチュエータ仕様を理解するとともに、必要に応じて新規のセンサやアクチュエータに対する要求仕様の定義を行う
- 制御システムのコントローラに求められる要求特性を定義し、コントローラのアーキテクチャ設計を行う
- 必要に応じてコントローラ内のH/W仕様とS/W仕様を分離し、それぞれの構成要素に対する要求仕様の定義を行う

3-4-2. プラントモデルエンジニア

【職種の説明】

制御システムにおける制御対象（プラント）をモデリングする技術者

* 機械設計を目的としたプラントをモデリングする技術者とは異なる。

【職種の役割】

- 制御システム特性をシミュレーション可能とするための制御対象（プラント）をモデリングする
- 制御システムに必要となるセンサやアクチュエータの動作を模擬できるモデリングを行う

3-4-3. 実装モデルエンジニア

【職種の説明】

制御機能モデルを元に、組み込み用のCソースが自動生成できるようにモデルを変更するエンジニア

【職種の役割】

- 制御機能モデルを元にS/W実装仕様書を作成し、実装可能なCソースを生成する制御実装モデルを作成する
- 単体テスト仕様書を作成し、単体テストを実施する
- 実装モデルの開発効率（コストと納期）と品質を確保する
- コントローラ内のH/W仕様とS/W仕様を分離し、それぞれの構成要素に対する実装仕様の定義を行う
- ROM、RAMの構成、I/F部の設定方法、入出力の定義
- 演算タイミング、割り込みの設定
- 制御機能モデルから求められる精度に見合った実装モデルを作成する

3-4-4. 適合エンジニア

【職種の説明】

与えられた制約範囲の中で制御システム全体の最適化を行う技術者

【職種の役割】

- 制御装置と制御対象を組み合わせたシステム全体を、最適に制御できるようにするために、開発プロジェクトにおける適合の要求事項と範囲に従って、制御定数の決定を行う。制御アルゴリズムはECUプログラムで定められており、適合対象はパラメータ値に限られている。
- テストベンチや試運転で必要となる電気／物理的信号を測定し、制御システムエンジニアとともに評価作業を行うことで、パラメータを解析・最適化する
- 適合データの管理を行う

3-4-5. 開発環境エンジニア

【職種の説明】

プロジェクトで使用するツール・設備等、開発環境の設計・構築、運用を担当する技術者

【職種の役割】

- 開発プロジェクトに対して、プロジェクト活動が円滑にかつ、効率的に行われるように各種支援業務を実施する
- 自動車制御システム開発において、以下の開発環境に関する、「計画」、「投資」、「調達」、「構築」、「運用」、「保守」、「トレーニング」などの業務を実施する
 - モデル作成・シミュレーションツール
 - コード生成ツール、コンパイラ、シミュレータ、デバッガ、エミュレータ
 - テストツール(静的解析、カバレッジ測定、テストデータ生成など)
 - ラピッドプロトタイピングツール、HILS
 - 適合ツール
 - プロジェクト管理、成果物管理、要件管理、変更管理、構成管理ツール
 - インフラ(サーバ、DBMS、ネットワーク)、各種計測用機材
- 運用方法や使用ツールのバージョン管理も含めた妥当性などについて評価と改善を実施する
- 標準規格適合などの要求に応じて、開発環境の認定が必要となる場合対応する

3-4-6. 職種とプロセスの関係

各職種とプロセスとの関係を下記表にて表す。

表 6 . 職種とプロセスの関係(参考例)

| プロセス | | 制御システム エンジニア | 適合 エンジニア | 実装 モデル エンジニア | プラ ント モデル エンジニア | 開発 環境 エンジニア |
|--------------|-------------|-----------------|-------------|--------------------|--------------------------|-------------------|
| 企画 | 分析 | | | | | |
| | モデル化 | | | | | |
| | ケーススタディ | | | | | |
| | 判断・決定 | | | | | |
| 機能設計 | 要求分析 | | | | | |
| | システム設計・モデル化 | | | | | |
| | 制御仕様検証 | | | | | |
| 実装 | 分析 | | | | | |
| | 実装モデル化 | | | | | |
| | コーディング | | | | | |
| | 結合 | | | | | |
| | 検証 | | | | | |
| 適合 | 実験計画 | | | | | |
| | 試験 | | | | | |
| | モデル化・最適化 | | | | | |
| | 検証 | | | | | |
| 制御システム 検証 | 検証計画 | | | | | |
| | 検証 | | | | | |
| | 判断・決定 | | | | | |
| プロセス改善 | 分析 | | | | | |
| | モデル化 | | | | | |
| | 運用 | | | | | |
| | 検証 | | | | | |

3-4-7. その他M B D 技術者と連携する職種について

自動車制御システム開発においてM B D 技術者と連携して業務を遂行する職種つまりM B D スキルの習得が推奨される職種を以下にあげる。

(1) 回路設計エンジニア

コントローラ、アクチュエータ、センサのH/W設計を行うエンジニア

(2) 機構設計エンジニア

アクチュエータ、センサを含む機械部品の設計を行うエンジニア

(3) S/W(P F : プラットフォーム)エンジニア

デバイスドライバ、リアルタイムOS、ミドルウェア及びA P I 等のS/Wプラットフォームを開発するS/Wエンジニア

(4) テストエンジニア

テスト仕様書に基づき、S/W、コントローラ、センサ、アクチュエータなどのテストを行うエンジニア

(5) 実機テストエンジニア

テスト仕様書に基づき、自動車または構成部品そのものでテストを行うエンジニア、テストドライバも含む

V. 付録

1. スキル定義例

各職種における開発スキルの定義例を下に記すので、参考としていただきたい。

1-1. 制御システムエンジニアスキル定義例

| プロセス | スキル | | 関連技術 | 説明 | Output | |
|--------------|-----------------|------------------|---------------------------------------|---|--|---|
| | 第一階層 | 第二階層 | | | | |
| 要求分析 | 構想把握 | システム概要把握 | | システム構想(企画書等)を元にシステム概要を図解する。図にはサブシステム単位の構成、サブシステム要素、制御で実現する項目(制御要素)等を記載する。 | | |
| | | システム概要レビュー | レビュー手法 | システム概要を元にシステムの構成要素をすべて含んだシステム構成図を作成。レビューにより、システム構成を確定する | システム構成図 | |
| | 要求獲得と調整 | 現行システムの問題点把握 | | 現行システムとの違いや現状の問題点を確認する | | |
| | | 新しい要求に関する情報収集と分析 | インタビュー手法 データ計測、FTA FMEA、ユースケース図 | ヒアリングやデータ計測などにより詳細な情報の収集やデータ分析を行う | | |
| | | ハードウェアの機能把握 | | 新しく採用する機構、センサ、アクチュエータの機能を調査し、内容を把握する | | |
| | 要求定義 | 要求内容の整理 | | サブシステム毎に要求内容を整理する | サブシステム毎の要求項目 | |
| | | 制御の機能分類と機能配置 | | サブシステムの機能要求を実現するため、サブシステム間にまたがる項目も含めて制御要素をすべて決定する | | |
| | | センサ特性把握 | | 各種センサの特性、動作範囲を確認する | | |
| | | アクチュエータ動作把握 | | アクチュエータの特性、動作原理を確認する | | |
| | 要求仕様レビュー | 制約条件把握 | | 動作、法規、コスト、ロバスト性上の制約条件を洗い出す | | |
| | | 制御システム構成の明確化 | FMEA | 要求項目から、実現すべき機能、非機能要求を確認し、実現するための振る舞いを考え、制御システム全体の構成を図解する | 制御システム構成図 | |
| | | 制御要求仕様書の作成 | | ハードウェアとソフトウェアのどちらで実現するかの分担と外部とのインターフェースを明確にする。次に各制御要素で実現すべき要求を明確にするため、制御要求仕様書を作成する。 | 個別制御要求仕様書 | |
| | システム設計 | 制御仕様設計 | システム要求事項との一貫性検討 | | システム要求事項との矛盾がないか、一貫性を検討する | |
| | | | 要求仕様書のレビューと評価 | DRBFM* | 要求仕様書のレビューを行い、要求仕様を確定する | レビュー後の個別制御要求仕様書 |
| | | | 制御構造設計 | Simulink/Stateflow | 制御要求仕様書を元に、制御システムの構造、構成要素抽出・整理、構成要素間のインターフェース設計を行う | 制御構造設計書 |
| 制御構造レビュー | | | DRBFM | 構造設計書をレビューし、実現方法・構造設計の妥当性を確認する | レビュー後の制御構造設計書 | |
| ロジックレビュー | | 制御モデル詳細設計 | Simulink/Stateflow | 個別の制御要素で実現すべき機能を詳細設計する | 制御モデル | |
| | | モデリングチェック | ガイドラインチェック | 制御モデルの可読性確認とガイドラインに沿ったモデルになっているかチェックする | | |
| | | 制御モデルレビュー | DRBFM | レビューを行い、指摘のあった箇所について制御モデルを修正する | 修正後の制御モデル | |
| | | 制御ロジック解説書作成 | | 制御仕様の理解を助けるため、制御ロジック解説書を作成する | ・ロジック解説書 | |
| | | 適合手順書作成 | | 適合の必要な場合は適合手順書作成 | ・適合手順書 | |
| | | 制御仕様検証 | モジュールテスト | 単体シミュレーションと動作確認 | Signal Builder, From workspace,SLDV Simulink/Model Verificationブロック Assertionブロック, Proof Objectiveブロック | シミュレーション検証、プロパティ検証などの方法(「JMAAB 制御仕様検証ガイドライン」参照)で制御モデルの動作確認を行う |
| 構造ベーステスト | | | | ホワイトボックステスト等によりカバレッジ計測する | | |
| 既存全体制御仕様との結合 | コード生成 RPT | | | 既存する他の制御仕様と結合する。ラビッドプロトコントローラを使い、バイパス手法またはフルパス手法で全体の制御仕様を構成する | | |
| 結合テスト | テスト実行と評価 | | MILS, SILS, HILS 実車テスト | クローズドループシミュレーションまたは、実機、実車テストにより不具合がないか動作を確認する | 修正後の制御モデル | |
| | 全制御仕様の結合 | | コード生成、RPT | 新規に作成した制御仕様をすべて集めて、既存の制御仕様と結合して全体の制御構造を構成する | | |
| | テスト実行、評価 | | SILS, HILS 実車テスト | クローズドループシミュレーションまたは、実機、実車テストにより不具合がないか動作を確認する | | |
| | 最終仕様レビューによる仕様確定 | | DRBFM | テストをすべて終了した後、最終的な制御仕様のレビューを行う | 最終版 ・制御モデル | |

1-2. プラントモデルスキル定義例

| プロセス | スキル | | 関連技術 | 説明 | Output |
|-------------------|------------------------------|--|--|--|--|
| | 第一階層 | 第二階層 | | | |
| 用途把握 | リアルタイム性・詳細度・必要な外部インターフェースの理解 | | MILS/SILS/HILS/PILS システム同定 制御設計 適合・検証 | 利用法に応じて、モデル形態、詳細度が異なるため、開発するプラントモデルの利用法を把握する。 | モデル形態モデル要求仕様書 (詳細度、リアルタイム実行、物理・実験モデル、制御設計用簡易モデル、など) |
| モデル対象解析・システムモデル設計 | 変数定義 | 入出力・外乱変数定義 内部変数定義 モニタ変数定義 | システム理論 ↑ HILS・モデル検証・適合 | システムに影響する入力と外部要因を解析する。 保存量・中間変数などの内部変数を定義する。多くの場合イタレーティブな作業になる。 モデル変数モニタの目的はモデル検証と運用に分かれる。 | 入出力・外乱変数の定義仕様書 モデル内部変数定義書 モニタ変数定義書 |
| | モデル要素の定義 | 物理・実験統合モデル: ・システムの要素分解 実験モデル: ・実験計画 ・システム同定 ・境界把握 | 力学・電気回路・流体・伝熱・化学反応などの基礎、システム同定・機械学習・統計理論、 常微分方程式論 偏微分方程式論 離散時間化 | ハードウェア構成要素毎に分解することが基本だが、必ずしもハードウェア構成と一致するとは限らない。 物理と実験モデルの組み合わせも決める必要がある。 | システム構成要素の定義 |
| | 要素間相互作用定義 | 非因果的モデリング: ・保存量交換 ・拘束記述 因果的モデリング: ・データフロー線図 ハイブリッドシステム: | ↑ | システムの振る舞いは外部から要素への影響と内部要素間の相互作用によって発展する。複雑なシステムでは階層的構造を持つが、実験モデルでは要素分解を行わないことが多い。 | モデル構造定義書 (モデル内部変数定義を含む) |
| | モデリング言語・ツール選択 | 常微分方程式の解法 | ・非因果的モデリング: Symscape・Modelica・VHDL-AMS ・因果的モデリング: Simulink | 一般に、異なる言語・ツールの併用やCo-simulationは容易ではない。可能な限る同一環境で行う方が無難である。 | モデリング言語・ツールの選択結果 |
| | 既存システムモデル調査 | | ↑ | 一般に、ベースとなる既存のシステムモデルを再利用できれば効率的だが、新規に作成の方が効率的な場合もある。再利用することにより拘束は避けられない。 | ・ベースとなるシステムモデルの入手、あるいは、新規開発の決定 ・改造・新規開発が必要なモデル要素の特定 |
| モデル要素開発 | 要素分解 | | モデリング技術 | 階層が異なるだけなので、システムモデル設計参照 | 階層が異なるだけなので、システムモデル設計参照 |
| | 制約定義 | | 微分・差分代数方程式 | ↑ | ↑ |
| | 相互作用定義 | | 力学・電気回路・流体・伝熱・化学反応などの基礎 | ↑ | ↑ |
| | システム同定 | | 関数近似理論・システム同定論・統計理論・機械学習・最適化・正則化・交差検証法・など | 実験モデルは理論体系が物理モデリングよりもしっかりしている。 | 実験モデル仕様書 システム同定仕様書 |
| | 要素モデルアセンブル | | 制御理論(安定論:モデル要素のアセンブルでフィードバック系を構成する) 微分方程式数値解法 stiff系 | 閉ループシステムの誤差は、それが十分小さく見えても、閉ループシステムでは大きな影響が現れることがある。場合によっては、安定性も失う。単に実験との誤差だけを評価しているだけでは不十分な場合もある。 | 要素モデル仕様書 要素モデル |
| システムモデリング | モデルアセンブリ システムモデルアセンブル | | ↑ | ↑ | システムモデル仕様書 システムモデル |
| | システムモデル調整 | | モデル簡易化(model order & parameter reduction,線形化など) 感度解析 システム同定・機械学習など | プラントモデリングでは、モデル要素が十分な精度であるとは限らないので、組み合わせれば目的とするシステムモデルができるとは限らず、LEGOの概念は使えない。モデル要素誤差が非常に大きな影響を持つ場合があり、システムモデル全体の調整が必要となることが多い。 | システムモデル調整仕様書 |
| | システムモデル簡易化 | | 数式簡易化(厳密・近似) 数値簡易化(近似) | システムモデルはパラメータスタディーや全運転範囲をカバーするシミュレーションなど膨大なシミュレーションを行うことが多いので、計算時間短縮から簡易化することがある。 | 簡易化仕様書 簡易システムモデル |
| モデル実行 | モデル実装 | | MILS/SILS/HILS/PILS 標準インターフェース | リアルタイムが要求される場合は、モデル開発環境とは別な実行環境に実装する必要がある。 | モデル実装仕様書 実装モデル |
| | 実行管理 | | モデル実行プロセス 実行プロセス定義言語 | 入力データのフィード、実行データの取得を可能な限り自動化する。 | モデル実行仕様書 実行管理プログラム |
| | データ変換 | | モデル実行の目的による。 | 可能な限り目的に直結するデータに変換する。 | 変換されたデータ |
| | 結果管理 | | モデル・入力・出力データベース | データ・モデル再利用を効率化する。 | データ管理情報 |
| モデル検証 | モデルフォーマット検証 | ガイドラインチェック | モデルガイドライン | 可読性を高めるために定められたガイドラインとの整合性を評価。 | ガイドラインと整合したモデル |
| | モデル実行結果検証 | | 力学・電気回路・流体・伝熱・化学反応などの基礎 非線形実験計画 評価条件データベース | 検証は学際的な技術とスキルが必要である。物理的に妥当な振る舞いか? 利用目的に沿った実行結果かなど総合的な判断が必要である。検証は可能性のある全運転範囲をカバーしなければならない。システムモデル調整と密接に関連している。 | 検証報告書 |
| モデルデータ管理 | モデル管理 | モデル再利用 構成管理 | モデルリポジトリ | モデリングを効率化し、モデルの再利用を容易にする。 登録対象は、システムモデルとモデル要素がある。モデルパラメータや検証情報も登録対象に含む。 | モデル管理情報 検証条件・データの登録情報 |
| | 検証条件・データ登録 プロセス管理 | | データベース | 検証条件とデータを再利用し易くする。 | 検証条件・データの登録情報 |
| | 物理法則登録 | | 力学・電気回路・流体・伝熱・化学反応などの基礎、評価条件データベース | 利用した物理法則を再利用し易くする。 | 物理法則登録情報 |

1-3. 実装モデルエンジニアスキル定義例

| プロセス | スキル | | 関連技術 | 説明 | Output |
|--------|-----------------|-------------------|--|--|-------------------|
| | 第一階層 | 第二階層 | | | |
| 分析 | 制御対象モデルの実現方法を決定 | 構造決定 | 制約条件: マイコン (ROM、RAM、スタック容量、実行時間)、通信 (容量、速度) | 要求分析を行い、システムに求められる要件を抽出し、ドキュメント化する。制御対象モデルの実現方法を決定する。制約条件を明確する。 | 制御仕様書 構造設計書 |
| 実装モデル化 | 制御モデルを実装モデルへと変換 | 最適化 (効率化、共通化など) | | 論理演算の最小化、実行時間・RAMの効率化、共通関数化、関数設計、ライブラリ設計 モデルの振る舞いに変化する変更はしない。 | 実装モデル |
| | | モデルの静的解析 | モデル記述ガイドライン CONTROL ALGORITHM MODELING GUIDELINES (JMAAB) | 実装モデルがモデル記述ガイドラインに合致しているかどうかを確認する。 | 静的解析結果 |
| | 単体検査 | 機能検査 | | 個々の機能の振る舞いに変化していないかを確認する。 | 機能検査結果 |
| コーディング | 実装モデルをCコードへ変換 | Cコード生成 | Cコード生成ツール | 実装モデルから、ツールを使用してCコードを生成する。 | Cコード |
| | Cコード静的解析 | コーディングルール チェック | | 生成されたCコードがコーディングルールに準拠しているかを確認する。 | |
| | 一貫性検証 | 実装モデルとCコードの一貫性検証 | Back-to-Backテスト | MC/DC100%となる検査パターンを使用し、Back-to-Backテストをおこない、一貫性検証をする。 | 一貫性検証結果 |
| 結合 | Cコードのオブジェクト化 | | コンパイラ | コンパイラを使用して、生成されたCコードをオブジェクト化する。 | オブジェクト モニタ用データ |
| | 単体検査 | 機能検査 | | 結合されたモジュールが制御仕様書で定義された機能や性能を満たしているかを確認する。 | 機能検査結果 |
| 検証 | | 一貫性検証 | Back-to-Backテスト | 実装モデルとオブジェクトの一致を確認する。 | 一貫性検証結果 |
| | 結合テスト、組合せテスト | | | 制御仕様で記述した機能及び前提とする制約条件を満足していることを確認するシステムレベルテスト (結合テスト、組合せテストともいう) を行う。 | 検証結果 |
| | テスト計画 | | | テスト手法の選択とスケジューリングする | テスト計画書 |
| | テスト設計 | | | ・仕様ベーステスト --機能要件/非機能要件/負荷 ・構造ベーステスト --データフローパス/モデル静的検証・動的検証/コード静的検証・動的検証 ・経験ベーステスト --過去不具合 ・レグレッションテスト ・等価性検証 (機能モデルとオブジェクトモジュール) | テスト設計書 |
| | テスト実行 | | | ・テスト記述 ・テスト環境 (PILS、HILS、信号試験機、実機) | テスト実行結果 |
| | テスト環境 | | PILS、HILS、SILS | テスト環境を構築する | 開発環境 |
| | テスト評価 | | | ・カバレッジ ・規模あたりテスト件数 ・規模あたり不具合件数 ・工数 | 評価結果、カバレッジ |

1-4. 適合エンジニアスキル定義例

| プロセス | スキル | | 関連技術 | 説明 | Output |
|-------|---|------|------------------------------|---|-----------|
| | 第一階層 | 第二階層 | | | |
| 実験計画 | 適合パラメータの抽出ができる | | 制御対象に関する知識 (ドメイン依存) 実験計画法 | 制御仕様の中から、制御対象・要求仕様のバリエーションに応じて適合すべきパラメータを選定する。 主に制御対象の諸元違いに起因するパラメータと、要求仕様で起因するパラメータがある | 適合パラメータ一覧 |
| | 初期適合ができる | | システム同定 最適化、最小2乗法 | 制御対象の諸元に起因するパラメータを設計値から求める、または制御対象の特性を同定し求める (初期適合とは、一般的にコントローラは制御対象に関するパラメータを使用し制御するため、制御対象の特性に応じたパラメータの設定を適合初期に行う) | 初期適合定数 |
| 試験 | 計測条件が設定できる | | 実験計画法 | 各種の性能指標と適合パラメータの関係を理解し、パラメータの最適値を決定するための試験条件を設定する。 | 試験計画 |
| | 計測器がつかえる | | 計測器知識 | 上記、試験に必要な計測器を使用しデータ計測を行う。 | 試験結果データ |
| モデリング | シミュレーションを使用した適合ができる (SILS、MILS、HILSを目的に応じて選択し使用できる) | | システム同定 境界モデリング | ・実機の無い段階での適合目処付け、故障モード模擬等、実機では困難な試験をシミュレーションで補完 ・実機とモデルの差異を理解しプロセスに応じてモデルの精度、手法を選択する。 | 適合定数値 |
| 最適化 | 適合パラメータの定数決定できる | | 最適化 最小2情報等 | 各種の性能指標を満足する範囲内で最適な適合定数を設定する | |

1-5. 開発環境エンジニアスキル定義例

| プロセス | スキル | | 関連技術 | 説明 | Output | |
|----------|---|---------------------------|---|--|--------------------------------|--|
| | 第一階層 | 第二階層 | | | | |
| 業務分析 | プロジェクト管理およびプロセス管理の要件定義 | プロジェクト管理およびプロセス管理ツールの要件定義 | プロジェクトマネジメント全般、PMBOK、ISO26262 | 現状業務および標準規格適合を分析しプロジェクト管理およびプロセス管理要件を定義する。 | プロジェクト管理およびプロセス管理ツールの要件定義(書) | |
| | 成果物管理および構成管理の要件定義 | 成果物管理および構成管理ツールの要件定義 | ソフトウェア開発技術全般、ISO/IEC12207、JISX0160、A-SPICE、ISO26262 | 現状業務および標準規格適合を分析し成果物管理、構成管理、要件管理、変更管理要件を定義する。 | 成果物管理および構成管理ツールの要件定義(書) | |
| | 要件管理および変更管理の要件定義 | 要件管理および変更管理ツールの要件定義 | ↑ | ↑ | 要件管理および変更管理ツールの要件定義 | |
| | MBD/ソフトウェアの開発環境の要件定義 | MBD/ソフトウェアの開発ツールの要件定義 | MBD技術およびソフトウェア開発技術全般 | 現状業務およびMBDプロセスを分析しMBD/ソフトウェアの開発環境構築のためのツール要件を定義する。 | MBD/ソフトウェアの開発環境構築の要件定義(書) | |
| 機能設計 | 事業継続計画(Business continuity Planning、BCP)の理解 | 災害による開発環境の影響分析 | 情報インフラ技術全般 | 天災または人災による開発環境への影響を分析しバックアップする要件を定義する。 | 情報インフラに対する要件定義(書) | |
| | 情報インフラとの調整 | IT部門との調整を行いインフラを設計 | ネットワーク、DB、冗長化 | 社内または社外ネットワーク環境における開発環境管理情報の取り扱いを設計する。 | ネットワークおよびサーバ構成および関連ソフトウェア設計(書) | |
| 実装 | MBD開発環境の選定・改善 | シミュレーション環境の構築 | MILS、SILS、RPC、PILS、HILS | 開発方式と各種ツールを理解・選定しそれらの開発環境を最適化する。 | MBD開発環境機能設計(書) | |
| | | プラントモデルとの連携 | プラントモデルの取扱い | プラントモデルエンジニアの支援を受けてシミュレーション環境を設計する。 | シミュレーション用プラントモデル | |
| | 組込みソフトウェア開発環境の選定・改善 | コード生成環境との連携 | コード生成、 | | | |
| | | プロセス管理ツール等との連携 | プロセス管理、構成管理、変更管理ツールなどのAPI | プロセス管理ツール等との効率的な連携を行うためインタフェースを設計する。 | MBD開発環境機能設計(書) | |
| 運用 | ネットワーク設定 | 実装環境との連携 | コンパイラ、シミュレータ、デバッグ、エミュレータ、 | ソフトウェア開発方式と各種ツールを理解・選定しそれらの開発環境を最適化する。 | ソフトウェア統合開発環境機能設計(書) | |
| | | 検証環境との連携 | 静的解析、カバレッジ測定、テストデータ生成 | | | |
| | DB構築 | プロセス管理ツール等との連携 | プロセス管理、構成管理、変更管理ツールなどのAPI | プロセス管理ツール等との効率的な連携を行うためインタフェースを設計する。 | ソフトウェア統合開発環境機能設計(書) | |
| | | 計測、適合 | ASAM、ECU通信インタフェースのハード/ソフト | 計測・適合方式を理解しI/Fを選定し計測・適合環境を最適化する。 | 計測、適合環境機能設計(書) | |
| ツール開発 | ネットワーク構築 | ネットワーク構築 | WAN、LAN、ファイアウォール | 情報システム部門と連携し各種開発環境をネットワークに組み込む。 | ネットワーク開発環境 | |
| | サーバー構築 | サーバー構築 | クラスタ、フェイルオーバー、ツール、ライセンスマネージャ、Linux、Windows Server | 情報システム部門と連携し各種開発環境のサーバーを設定する。 | サーバー | |
| 情報システム運用 | DB構築 | DB構築 | RDB、クラスタ、フェイルオーバー、SQL、JDBC | 情報システム部門と連携しDBシステムを構築する。 | 開発環境用DB | |
| | ツール開発 | 一般的ソフトウェア開発 | 各種言語(JAVA、mなど)、統合開発環境(Eclipseなど) | 各種システム、ツールのAPIを利用して各種システム、ツールをカスタマイズする。 | カスタマイズされた各種システム、ツール | |
| トレーニング | 情報システム運用 | 情報システム運用 | | 情報システム部門と連携し障害対応、バックアップ(障害時リストア)を実施する。 | 運用体制、手順(書) | |
| | トレーニング | トレーニング | | 各種開発環境のトレーニングを実施する。 | トレーニング資料 | |
| コンサル | コンサル | コンサル | | ユーザに対してコンサルを実施する。 | コンサル資料 | |

2. 教育用スキル分解例

前述したように、教育を目的とする場合は教育可能な技術や知識まで分解する必要がある。前項で各職種についてスキル定義例をあげているが、その中から各職種で1項目、教育のために分解した例を下に記す。

2-1. 制御システムエンジニアスキル分解例

| スキル項目 | スキルレベル1 (初級:読める) | スキルレベル2 (中級:使える) | スキルレベル3 (上級:分析・解析・活用できる) |
|-----------|---|---|--|
| 制御モデル詳細設計 | 既存の制御仕様(simulink)を開き、Simulinkの各ブロックの動作内容が理解できること。 | Simulinkブロックを使い、状態方程式を含んだ制御仕様を作成し、それを実行して、オーバーシュートやチャタリングなどの異常がないことを確認することができる。 | 状態方程式などを含んだ複数の制御仕様を結合し、制御対象モデルと組み合わせさせてMILS等を行い、複数の制御項目を最適に制御できる状態を作り出すことができる。 |

2-2. プラントモデルスキル分解例

| スキル項目 | スキルレベル1 (初級:読める) | スキルレベル2 (中級:使える) | スキルレベル3 (上級:分析・解析・活用できる) |
|-----------|--------------------------------------|----------------------------------|----------------------------------|
| システムの要素分解 | 既存モデルの要素分解、変数定義、相互作用の定義を理解でき、変更ができる。 | 新規モデルの要素分解、変数定義、相互作用を定義することができる。 | 物理・実験モデルとその統合したモデル変数を定義することができる。 |

2-3. 実装モデルエンジニアスキル分解例

| スキル項目 | スキルレベル1 (初級:読める) | スキルレベル2 (中級:使える) | スキルレベル3 (上級:分析・解析・活用できる) |
|----------------|--|--|----------------------------------|
| 最適化(効率化、共通化など) | 制御モデルを実行し出力波形を確認することができる。 実装モデルへ変換するために必要なモデリングスキル(論議演算の最小化、実行時間・RAMの効率化、共通関数化、関数設計、ライブラリ設計など)を知っている。 | 制御モデルで実現されているモデルの仕様が理解できる。 実装モデルへ変換することができる。 (モデルの振る舞いが変化する変更はしない) | 要求の目的を理解して、ロジックが最適化されたモデルの提案できる。 |

2-4. 適合エンジニアスキル分解例

| スキル項目 | スキルレベル1 (初級:読める) | スキルレベル2 (中級:使える) | スキルレベル3 (上級:分析・解析・活用できる) |
|---------------|-----------------------------|--|-----------------------------|
| システム同定:同定式の推定 | 処理済みの計測データを用いて、規定の次数で同定できる。 | 除外すべきデータを正しく認識し、使用するデータを限定、正しいデータだけに加工できる。 | システムに最適な次数を検討し、決定できる。 |

2-5. 開発環境エンジニアスキル分解例

| スキル項目 | スキルレベル1 (初級:読める) | スキルレベル2 (中級:使える) | スキルレベル3 (上級:分析・解析・活用できる) |
|---------------|-----------------------------|-------------------------------------|--|
| MBD開発環境の選定・改善 | 既存のツールチェーンを開発対象に適用することができる。 | ユーザーの改善要望を把握し既存のツールチェーンを改善することができる。 | 新規のツールを調査しツールチェーンに組み込むことができる。また、ベンダーに対してカスタマイズを要求する。 |

3. 活用事例

3-1. 開発環境エンジニアのための Simulink API スキル定義

表 7. 開発環境エンジニアのための SIMULINK API スキル定義

| やりたい事 | 業務レベル | 設計スキル項目 | | | | | | | | | | | | | | | | | | | | | |
|---|--|----------|-------------|---------|-------|------|--------------|---------|-------|---------------|-------------|----------|-------|----------|------------|--------------|-------|-----------------|-----------|------------|---|---|---|
| | | プログラミング論 | MATLABのデータ型 | ファイル入出力 | 文字列操作 | WS操作 | Simulinkブロック | モデル情報検索 | モデル構築 | Stateflow情報検索 | Stateflow構築 | シミュレーション | GUI作成 | グラフィック作成 | レポートジェネレータ | Simulink V&V | コード生成 | S-function及びTLC | マスク・ライブラリ | コンフィグレーション | | | |
| コード生成の為に、モデル内部情報検索を行い、ブロックプロパティを読み出し、設定したい規定プロパティに変化させる | レベル1 | 1 | | | | | 1 | 1 | | | | | | | | | | | | | | | |
| | レベル2 (特定のプロパティに関して作業できる) | 1 | | | | | 1 | 1 | 2 | | | | | | | | | | | | | | |
| | レベル3 (プロパティの制限なし) | 2 | | | | | 2 | 2 | 3 | | | | | | | | | | | | | | |
| | レベル4 (目的に対して最適な手法を選択できる) | 3 | | | | | 3 | 3 | 4 | | | | | | | | | | | | | | |
| 自動的に、設定に従って、モデルを構築する。 | レベル1 (新規作成、置換などの単純作業) | 1 | | | | | 1 | 1 | 2 | | | | | | | | | | | | | | |
| | レベル3 (ブロック挿入などのアルゴリズムと併用する作業) | 2 | | | | | 2 | 2 | 3 | | | | | | | | | | | | | | |
| | レベル4 (目的に対して最適な手法を選択できる) | 3 | | | | | 3 | 3 | 4 | | | | | | | | | | | | | | |
| | Stateflow内部情報検索し、日本語の概要を書き込んだり、スペースを入れてテキスト位置をそろえる。 | レベル1 | | | | | | | | | | | | | | | | | | | | | |
| モデルアドバイザーのチェックに基づき、Stateflowオブジェクトを自動的に構築・修正する。特定のパターンに沿った並び替 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 (特定のプロパティに関して作業できる) | 1 | | | | | 1 | | | 2 | | | | | | | | | | | | | |
| | レベル3 (プロパティの制限なし) | 2 | | | | | 2 | | | 3 | | | | | | | | | | | | | |
| | レベル4 (目的に対して最適な手法を選択できる) | 3 | | | | | 3 | | | 4 | | | | | | | | | | | | | |
| 任意のテストパターンでシミュレーション実行する。 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル3 (実用mファイルを作成できる) | 3 | 2 | 2 | | 3 | | | | | 4 | | | | | | | | | | 2 | | |
| | レベル4 | | | | | | | | | | | | | | | | | | | | | | |
| Simulink実行結果のログングデータの可視化 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 (2Dの可視化ができる。1画面に1波形) | 2 | 2 | 2 | | 2 | | | | 2 | | 1 | | | | | | | | | | 2 | |
| | レベル3 (3Dの可視化ができ、ハンドグラフィクス機能が使える) | 3 | 2 | 2 | | 2 | | | | 3 | | 2 | | | | | | | | | | 2 | |
| | レベル4 (目的に対して最適な手法を選択できる) | 3 | 2 | 3 | | 3 | | | | 4 | | 2 | | | | | | | | | | 2 | |
| モデル実行時に、リアルタイム表示されるGUIを設計し、MILS環境を向上させる。 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル3 (必要なリアルタイムGUIを構築できる) | 2 | | | | 3 | | | | 3 | 3 | | | | | | | | | | | 2 | |
| | レベル4 (目的に対して最適な手法を選択できる) | 2 | | | | 3 | | | | 3 | 4 | | | | | | | | | | | 2 | |
| 作ったツールを使いやすい環境を提供し、モデル設計者の作業工数を削減する。 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 (GUIDEを使った場合作る) | 2 | | 2 | 2 | 2 | | | | | 2 | | | | | | | | | | | | |
| | レベル3 (プログラミンで作れる。ユーザーデータプロパティが使いこなせる) | 3 | | 2 | 3 | 3 | | | | | 3 | | | | | | | | | | | | |
| | レベル4 (タブ、マルチウィンドウなどの使い分けができる) | 3 | | 2 | 3 | 3 | | | | | 4 | | | | | | | | | | | | |
| Simulinkの情報を取得してレポート (ワークスペース情報のリスト化) | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 (テキスト、EXCELに出力できる) | 2 | 2 | 2 | 2 | 2 | | 2 | 2 | 3 | | | | 2 | | | | | | | | 2 | |
| | レベル3 (見栄えを良くした出力ができる。レポートジェネレータを活用したり、モデルのキャプチャなどの画像情報も出力できる) | 3 | 2 | 2 | 3 | 3 | | 3 | 3 | 3 | 3 | | | 3 | | | | | | | | 2 | |
| | レベル4 (目的に対して最適な手法を選択できる) | 3 | 2 | 3 | 3 | 3 | | 3 | 3 | 3 | 3 | | | 4 | | | | | | | | 2 | |
| 複数のテストシナリオでのトータルカバレッジ取得 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル3 (必要な手順を自動化できる) | 3 | | 3 | 3 | | | | | 3 | | | | 3 | | | | | | | | | |
| | レベル4 | | | | | | | | | | | | | | | | | | | | | | |
| オリジナル自動チェック作成 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 (1関数で完成するチェック/自動修正であれば作成できる) | 2 | 2 | | 2 | 2 | | 2 | 2 | 2 | | | | | | | | | | | | 3 | |
| | レベル3 (機能が複合した(例えばブロックごとにオプションが異なるような)チェック/自動修正を作成し組み込むことができる。) | 3 | 2 | | 3 | 3 | | 3 | 3 | 3 | 2 | | | | | | | | | | | | 4 |
| | レベル4 (チェックコードのわかりやすさ、実行効率も考えて作成できる) | 4 | 2 | | 3 | 3 | | 4 | 3 | 3 | 2 | | | | | | | | | | | | 4 |
| オリジナルの標準的に使用できるマスクサブシステムを設計し、社内で活用できるようにする。 | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 (特定ブロックのみであれば作成できる) | 1 | | | | | | | | 2 | | | | | | | | | | | | | 2 |
| | レベル3 (ブロックの制限なし) | 2 | | | | | | | | 3 | | | | | | | | | | | | | 3 |
| | レベル4 (カスタムブロックのわかりやすさ、実行効率も考えて作成できる) | 2 | | | | | | | | 3 | | | | | | | | | | | | | 4 |
| オートコード用カスタムブロックの作成とライブラリ化 (S-function作成、TLC編集) | レベル1 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル2 | | | | | | | | | | | | | | | | | | | | | | |
| | レベル3 (LGT、S-functionBuilderを使えば自力で作成できる) | 2 | 2 | | | | | 3 | | | | | | | | | 3 | 3 | | | | 3 | |
| | レベル4 (ハンドコードでのS-function作成、TLCの編集もでき、目的に応じて最適な手法を選択できる) | 3 | 2 | | | | | 4 | | | | | | | | | 4 | 4 | | | | | 4 |

「表 6. 開発環境エンジニアのための Simulink API スキル定義」は、JMAAB Simulink API - WG で作成された。「Simulink API」を使うというスキルを教育レベルまで分解した例である。

3-2. W社推進プロセス事例

次に、MBDエンジニアに対する取り組み例を紹介する。

目的：求められる MBD エンジニア像を明確化し、そのステップアップ方法を作成する。

新人に対して、設計する為に必要な知識、経験を 1 か月間で提供するカリキュラムを開発。このカリキュラムによって、短期間でレベル 1 を超え、レベル 2 の途中まで育成できることが確認できた。

さらに、研修なしに対して、初歩のままで行った作業工数と、研修後の作業効率に格段の差があり、初期の開発工数の削減分で、研修期間分が削減されることが解った。

さらに、品質の高いモデルが設計できるので、新人に対して必ず研修を実施することにした。

運用を続けると、問題点が出てきた。

中堅社員向けのカリキュラムは、新人 4 週間に対して 2 週間まで圧縮しているが、人によってはさらに半減できることが解っている。しかし、教育スタイルが 3 名以上でのグループディスカッションを取り入れているため、どうしてもグループ内にレベルの違いがあると、優秀な人の時間が余ってしまう傾向がある。

当然、新人同士でもレベルの差が大きく、標準よりも優秀な人は時間が余ってしまう。

事前に、レベルの計測ができれば、圧縮した講習を受けられるようにグループ設定時に配慮が出来たかも知れないが、現状は、研修後半になってその差が明確となるので、一人一人にあった最適な教育の提供が難しく、中堅社員向けの教育が停滞している。

コメント：

開発スキル毎に講座を提供できるなら、一人一人に向けた最適な教育カリキュラムの選択が可能だろう。

しかし、開発スキルの習得は、一つの技術要素に対して、複数の開発スキルを組み合わせで初めて設計が可能であるケースが多い。そのため、一連の成果物を作り上げるコースに対して、一部分を切り出すレベル別のコース設計は非常に難しい。

3-3. X社推進プロセス事例

以下は、ETSSを会社全体で取り組んだ事例である

(1) ボトムアップ作戦<MBD+ソフト開発+適合業務部門で開始>

(a) MBD+ソフト開発+適合業務部門<初回開始>

該当部門に属するJMAABメンバーであった担当者がETSS-JMAABをベースに該当部門長の承認を得て以下プロセスで推進した。

- ① 部方針としてオーソライズ
- ② X社MBDエンジニアスキル標準の作成(図12、図13参照)
- ③ スキル診断シートの作成
- ④ 説明会実施
- ⑤ 不定期にスキル診断実施
- ⑥ 活用は、本人及び本人上長それぞれで個人活用に留まる。
- ⑦ 全社報告実施

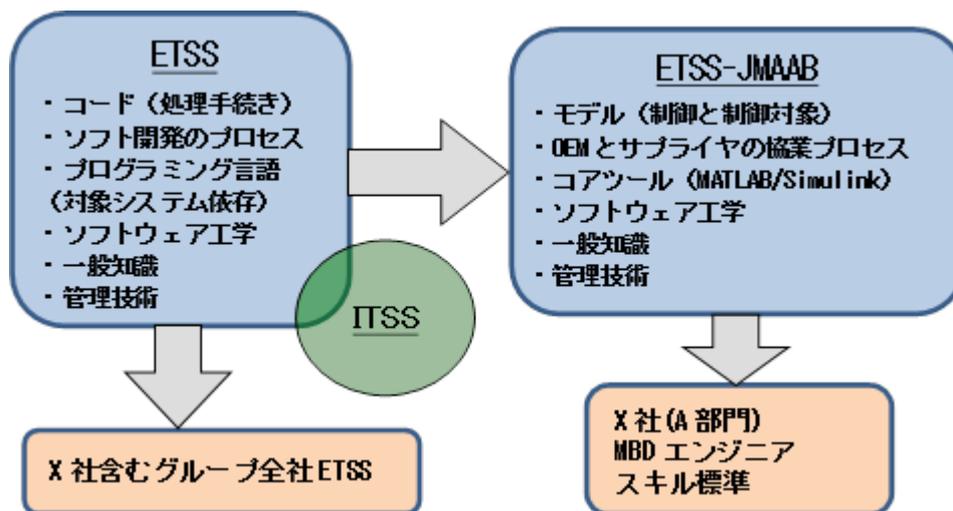


図 12. X社A部門MBDエンジニアスキル標準の位置づけ

(b) ソフト開発部門<ソフト開発業務部門に横展会>

上記(a)の状況を見た該当部門長が、以下プロセスを推進した。

- ① (a)の事例紹介
- ② 部方針としてオーソライズ
- ③ 部門スキル基準の策定、スキル診断シートの作成
- ④ 説明会実施
- ⑤ 不定期にスキル診断実施
- ⑥ 活用は、本人及び本人上長それぞれで個人活用に留まる。

(2) トップダウン作戦<コーポレート部門による全社展開/リスタート>

経営幹部から社員の技術者レベルに関わる課題が、品質問題に発展する可能性を指摘された。そこで、コーポレート部門が、上記(a)、(b)の事例を参考に見直し全社該当部門で実施した。結果活用は、今後の課題である。

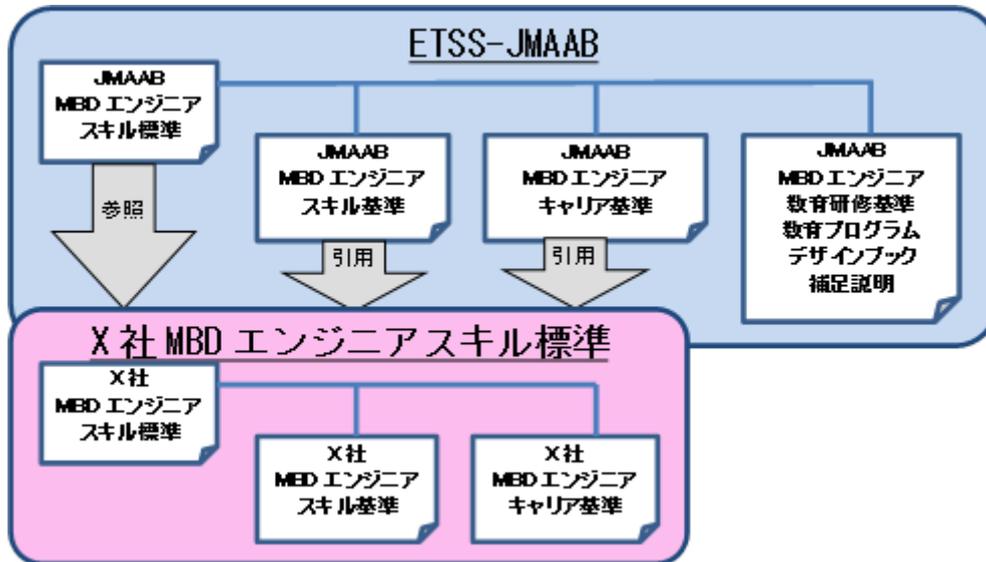


図 13. X社A部門MBDエンジニアスキル標準の作成

コメント：

第1ステップとして人材が見えるようにするという、スキルマップの一つの役割は達成している。

しかし、目指す人材像を定義し、個々のエンジニアをどうやって育成していくのか、計画に落とせなければ、スキルマップを作って終わりになってしまう事例。

多くの会社がスキルマップの記載はできているが、活用に進めない。

スキルマップを作るまでと、その先のステップに進むところに大きな壁がある。

それは、スキルマップのスキルと人材育成が直接的にリンクできていないからである。スキルマップのスキルを身に付けるために、何をすれば良いのか、分析ができていないからである。

あるいは、既存の教育カリキュラムを受けることで、スキルマップのどのスキルがどのレベルに上がるのか、講座以外の演習経験をどうやって受講するのか、どうやって評価するのかが解っていないからである。

自社でやらなければならない技術要素スキルはOJTで育てるしかない。そこで使われる開発スキルは外部の教育コンテンツを上手に活用し、育成プログラムを作り、運用すべきであろう。

3-4. Y社推進プロセス事例

以下も、取り組み事例である。

(1) 対象領域

量産組込み・開発環境構築開発者

(2) 育成フェーズ

以下、(a)～(c)の3つのフェーズで育成する。

(a) ベースローテーション

ETSS S/Wエンジニアに相当。組込みプラットフォーム開発(PF)と組込みアプリケーション開発(AP)のエントリーレベル(L1～L2)・ミドルレベル(L3～L4)を対象とし、入社から5、6年目までの期間で組込みS/W技術者としての基礎知識を習得する。



ベースローテーションの目標スキルを達成すると、スペシャリスト育成ローテーションへ移行する。ただし、以下を満たすこと。

- ・全項目レベル2 以上
- ・スペシャリスト育成ローテーションの中、進むべき専門領域が決まっている

(b) スペシャリスト育成ローテーション

システムアーキテクト(SA)、開発環境エンジニア、開発プロセス改善、組込みアプリケーション開発(AP)、組込みプラットフォーム開発(PF)、QAスペシャリスト(QA)のハイレベル(L5～L7)に相当する。5、6年目から10年間ほどを対



象とし、専門領域のスキルを極める。

(c) マネージャ育成

プロジェクトマネージャ、ブリッジエンジニアのミドル・ハイレベル（L3～L7）に相当する。プロジェクトを統率できる人材を育成する。



(3) 評価

半期に一度実施する。

コメント：

育成のステップを経験年数で表現している。

これに対して、ETSS-JMAABでは、必要なスキルが習得出来たら次のステップに進むべきという手法を提案している。

スキルをきちんと測ることが難しいため、経験年数でスキルを定義しがちであるが、経験年数ではなく、スキルを数値で表現できるようにすべきです。

3-5. Z社推進プロセス事例

ソフトウェア開発部署での事例

ETSSの発行当時からスキルマップによる技術者のスキル見える化に着手。

技術力向上の目標を掲げ、数年のトライにより、技術要素スキルの体系が完成。

個人のスキルアップにスキルマップを活用し、技術要素毎に年間目標を立て、定期的に講習会を実施、スキルアップを着実に実施した。

取り組みから、数年が経過し、当時新入社員だった人たちもリーダー格に成長した。

一見、順調にスキルアップの取り組みが定着したように思えた。

そのような中でソフトウェア設計監査を受けた時に発覚した衝撃の事実。

当然、教育に対しての取り組みだけではなく、並行してプロセスに対しても改革を実施していた。各プロセスでやるべきことは明確である。現に製品の不具合は確実に減少していた。

そこで、自信満々で設計工程の監査を受けて解った事実。

実は、なぜそのプロセスがあるのか、なぜそのプロセスで定義された作業を行わなければならないのか、知らなかった。しかもリーダー格の人たちですら。

これは、なぜか。

技術要素スキルマップと別に、開発スキルの定義が必要であった。

ソフトウェアを設計するのだから、当然知っておくべきソフトウェア工学、その他設計に使用する基礎技術がある。

これらについては、プロセスに定義があるし、OJTで使い方、やり方を聞いているので、使いこなしてはいたが、本質を知らなかった。

例えば、なぜC0、C1カバレッジのテストをこの工程で行うのか？

C0、C1カバレッジを取る為に、なぜこの手法でテストケースを検討するのか？

これらを知らずに、「プロセスに定義されているから、あるいは標準手順書に書いてあるからやってみました。」であった。

つまり、ソフトウェア工学をきちんと理解していなかったことが解った。

現在、開発スキルの定義とスキル習得に対する取り組みを実施中。

コメント：

技術要素スキルマップさえ作れば、人材の評価と、育成結果を見ることは可能である。最も製品に直結し、結果が解り易いからである。しかし、それが行き過ぎると基礎をおろそかにしてしまい、エンジニアの質の低下を招く結果となる。

学術的な基礎知識が無くても、使い方さえ解れば、結果は出せる。全ての人が基礎知識を完璧に習得し、理解する必要は無いが、キーマンとなる存在は必要である。

目標と直接結びつかなくとも、足元の基礎のスキルは重要である。

4. 教育に関する注意点

ETSS-JMAABや、その元となるETSSを用いたエンジニア育成手法について、注意点を記載する。

まず、これらの手法を取り入れる目的をはっきりさせる必要がある。そして目的に応じたキャリア定義、スキル定義を行わなければならない。

人材の評価を行うだけなら、技術要素スキルを粗めに定義し、スキルを定義、マップを作成すれば完了である。しかし、これを人材育成に使うとなると、非常に難しい取り組みになる。

育成を行う場合は、まず初めに、育成目標を明確に定義する必要がある。

E T S S はエキスパートの育成はある意味で対象にしてない。なぜなら、優秀な人材を育てる方法と、標準的な人材を大量に育成する方法は、手法から異なるからである。エキスパートの育成は、みなさんご存知のオリンピック選手の育成が参考になる。オリンピック選手は、まず光る人材を見つけるところから始まる。つまり教育する対象を絞り込んでいるのである。そして逸材を見つけたら、集中的にマンツーマンに近い形で育成を行う。指導する側も、指導される側も選び抜かれた人材で教育を行う。しかし、ここで記載しているスキルマップを用いた人材育成は、限られたエキスパートを育てるずっと前の段階を想定している。選び抜く前段階の標準的な教育だと考える。したがって、目標とする人材像もエキスパートのずいぶん手前にいる、標準的なエンジニアあるいは、新人と標準的なエンジニアの間ぐらいを想定する。最終的な育成後ではなく、その手前の段階と考えた方が良い。一人前の立派なエンジニアを育てるには 10 年以上の歳月が必要であり、そのような長い期間の育成カリキュラムを全て用意することは実質的には不可能である。できる範囲で、取り組まないと目標到達まで長すぎて途中で挫折してしまいかねない。

さて、目標とするエンジニアを明確にすると言う作業はさらに続く。ターゲットとなる目標のキャリアを定義しなければならない。キャリアの定義のためには、プロセスの定義、プロセスごとの成果物の定義が必要である。プロセスと、会社の組織が関係していることも重要である。自社にあったプロセスの定義がきちんとできていれば組織が紐づく。組織と無関係のプロセスができていれば、その定義は間違っていると考えても良いであろう。良い製品が既に設計できているならば、その組織もある程度成熟された現実に沿った形で形成されているはずである。作業の流れだけでは工程の切れ目の定義が難しいので、組織を参考にすると良いであろう。

プロセスが定義され、キャリアが定義できたら、スキルの分解を行う。プロセスごとの成果物からスキルを定義する。開発スキルの多くは成果物を設計するための技能である。また、成果物そのものが、複数の工程で成熟していく使われ方をしているようなら、それも開発スキルとして抽出する。

技術要素、開発スキルが定義されたら、それぞれのスキルレベル定義と、キャリアレベルの定義、キャリアレベルごとのスキル目標値を決める。

最後にスキルレベルをアップさせるための教育コンテンツを検討し、育成計画を立てる。

手順としては、上記のようになるが、実際に技術要素スキルと、開発スキルを分離するのは非常に大変な作業である。開発スキルと言っても、必ずドメインの定義が必要になり、ある程度技術要素スキルと関連が出てくる。

最終目標が「仕事ができる」ことであれば、教育題材は業務と結びついて当然である。学術的に正しい必要はない。実践できることこそが大切なのである。つまり技術要素スキルと、開発スキルと完全な分離は不可能である。しかしそういったことは、ある程度割り切らねばならない。抜け漏れなくスキルを抽出し、教育する手法を考えることが大切である。

技術要素、開発スキルを教育コンテンツへ落とし込むには、実際の成果物の作り方を分析して、何を教える必要があるのか、検討する必要がある。表面的な現象に注目し、本来教えるべきことを見逃さないようにしなければならない。

例えば、MBDができる方々の特徴として制御理論を勉強していた人が多かったとしても、万人に制御理論を教える必要はない。制御理論を知っている人は、制御工学を勉強している。制御工学では、システム同定を行い、プラントモデルを設計し、MILSと言うシミュレーションを用いた検証を行い、そして実機へ実装するという制御手法の王道を身に付けている。制御理論を勉強した人がMBDで力を発揮するのは、制御理論を知っているからではなく、おそらく制御工学を身に付けている、あるいは制御工学が得意とする制御開発プロセスを理解しているからである可能性が高い。

そもそも、何を教育すると目的とするエンジニアに近づくのか、それは教育担当者がきちんと考えるべきである。

そして、モデルベース開発の重要なポイントは、シミュレーションを活用することである。シミュレーションを活用できないプロセスを作り、エンジニアに仕事を押し付けて教育しても、MBDを採用しても工数が減らない、人材のレベルが上がらないと言うのは間違いである。

教育以前にきちんとした制御開発プロセスが採用できているのか、上位者視点で見つめる必要がある。

「教育担当者は、教育のことだけを考える」、これはエンジニアの本質から離れている。MBDエンジニアの教育を検討するのもエンジニアであり、教育というシステムをモデリングし思い通りにコントロール（企画・設計・運用・評価）しなければいけないのである。常に上位者視点で、物事を考え、それを指導し、受け継いでいかなければ、良い製品の開発はできない。何事も真剣に、全力で取り組む必要がある。

それが、最終的にユーザが手にする良い製品に繋がる。

教育の最終目標も、良い人材を育成することではなく、より良い社会に貢献することであることをエンジニアとして忘れてはいけない。エンジニアの究極目標は、常にお客様のため、あるいは社会のために貢献することなのである。

5. プロジェクトで活用する際の注意点

プロジェクトでの活用する場合は、教育とは少し異なる。

プロジェクトチームの設立に対して、スキルマップやキャリア基準マップを活用し、必要とするスキルを持った人たちを集め、チーム全体で各分野の技術要素スキルがカバーされるよう、チームを決定する場合にスキルマップやキャリア基準を活用できる。

誤解された使い方として、レベル不足を人数で埋められると勘違いする使い方である。

特定のスキルに関しては、レベルとは、ある規定時間内の作業を示すかも知れない。レベルの高い人ほど同じ作業を早くできる。それはそのようなケースもあるだろうが、ことキャリア基準に関しては、作業効率ではなく、能力に対する定義である。基準を満たしている事と、満たしていないことは、できるか、できないかである。つまりあるレベルとあるレベルを比較すると何かの設計が、できるかできないかの違いがある。

そのようなケースにおいては、レベル不足の人を何人入れようが、不足した能力は人数では埋まらなないと考えるべきである。もちろん、能力の高い人が既にいれば、人数を増やすことでそのエンジニアの空き時間を作り、1人が必要な部分だけに集中することで、2人分の仕事をする事は可能だろう。しかしながら1と0には大きな違いがあることを認識すべきである。

「そんなことはない。現実当社では人数を増やすことで今まで対応できている。」と言う会社があったとする。それは、本当だろうか。そういった会社は、「スキル認定が間違っているのではないか。」と疑うべきである。つまり、エンジニアのレベルを能力で判定せず、経験年数で評価していないか。

非凡な才能というものは、経験年数では評価できない。そのような会社では、若手が大きなプロジェクトを成功させ、エキスパートと言う名のご老人が関係するプロジェクトは、暗礁に乗り上げているのではないか。

このような運用のミスを犯さないためにも、正しく評価が行えるスキルの定義や能力の判断が必要である。

また真のエキスパートは、スキルマップでは正しい評価ができないかも知れない。その点についても認識しておくべきである。

標準化した規格や基準と言ったものは、あくまで標準化した普通の人たちや、それ以下を区別することはできても、非凡な人を評価することはできない可能性が高い。

そして、非凡な人材は、凡人を何人集めて教育しても非凡な人を超えることはない。

非凡な人は、その能力を発現の兆候を早めに見つけて、意図的に成長させさるよう、オリンピック強化選手なみに特別選定して育成すべきである。

そして、上司は上司も含め、凡人は非凡な人を認めないことを理解すべきである。なぜなら自分にそれ（能力）がないことを認めたくないからである。

それを理解した上で、非凡な才能がある人は非凡な才能がある人が見つけ出し、普通の人とは異なる手法で評価できる仕組みが必要である。

そして、そのような仕組みを持ってこそ、運用が可能であるべきと考えなければならない。

6. 用語解説

ここでは、ETSS-JMAABにおいて特徴的に使用され、またキーとなる用語の解説を行う。

6-1. スキル、技術

ETSS-JMAABでは、「スキル」、「技術」に対して、それぞれ明確に定義することでその違いを峻別し、使い分けを行っている。

6-1-1. 技術

ETSS-JMAABにおける技術とは、要求に対する結果を導くために、経済原則（コスト条件など）を満足するように手順化・体系化された再現可能な工程（プロセス）のことである。

技術は明示的に知識化することが可能であり、文書や製品あるいは教育などの手段によって不特定多数の人に伝達することが可能であるという特性を持つ。つまり技術は、客観的に検証できる工学として成立するものである。

6-1-2. 技術力

「あの会社は技術力がある。」といったように、一般的に技術力という用語が使われるが、ETSS-JMAABにおける技術力とは、技術そのものを、「開発」「改善」「革新」「実現」でできる能力である。

技術は要求から結果を導き出すための工程（プロセス）であり、技術力とは、その工程自体を新たに創出したり改善改良したりできる能力であり、いわゆる「技術革新（イノベーション）」を推進する力ともいえる。

6-1-3. スキル（技能）

ETSS-JMAABにおけるスキルとは、要求に対する結果を導く技術全体あるいは技術の一部（サブ工程）を実行する個人の作業遂行能力である。

スキルは個人に依存するものであり、言語化あるいは機械化された知識だけでは伝達できない特性がある。スキルは技術を活用する訓練を含む経験の積み重ねから個人の中に熟成されていくものである。

つまり技術を活用できるスキルを伝承させるためには、個人から個人への繰り返しによる訓練とそのための環境を提供する必要がある。

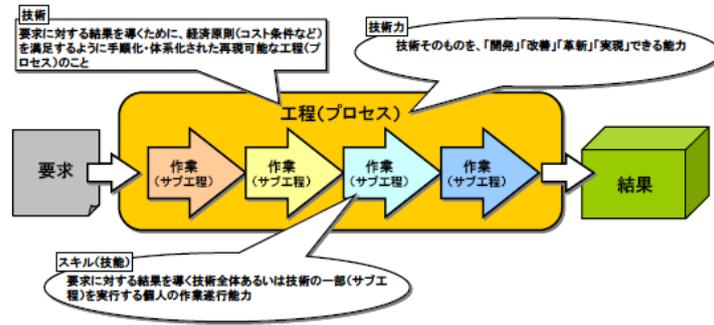


図 14. スキル(技能)と技術力

6-2. 技術者と技能者

ETSS-JMAABでは、技術者と技能者を以下のように定義している。

【技術者 (Engineer)】

要求に対する結果を導くための工程（プロセス）に対して、技術、技能（スキル）、経験などを駆使して、新たな工夫による実現、改善、革新ができる技術力を持った人材。

【技能者 (Technician)】

工程（プロセス）あるいはサブ工程において必要となる作業を提供できる人材。

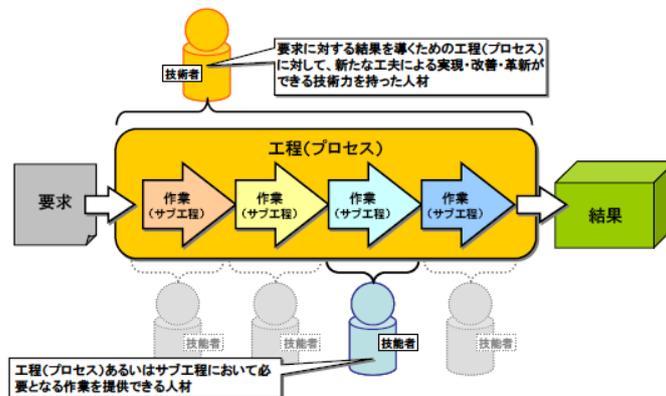


図 15. 技術者と技能者

製品開発実務において、技術として安定した既存の工程や作業（サブ工程）は技能者でも実現可能である。しかし、より高い要求品質に応えるために既存技術を改善したり、技術革新（イノベーション）を起こし組織や業界をリードしたりするのは技術者である。ただし、両者を二律背反的な人材と見るのではなく、一人の人材が技術者の顔と技能者の顔を持つと考えるべきである。

開発スキルは、技能者の側面に注目したスキル表現が行われる。しかし技術者は技能を使えるだけではない、技能を使いこなし新たな製品を開発できることが必要である。

6-3. スキル分布

ETSS-JMAABでは、その技術項目ごとにその技術を活用できる習熟度を6段階で評価する。したがって、スキルの評価は分布となることに注意されたい。スキルは分布で表現されるものであって、1個の値で表現されるものではない。

6-4. キャリア基準

ETSS-JMAABでは、ある職種で実務を行うために必要な経験や能力を数値で表現し、基準として表現したもの。職種ごとに全てのスキルをまとめて表現する。

6-5. キャリア

プロフェッショナルとか経験と称され、職種と同義語として用いられることが多いが、職種のどのレベルにあるかまでを総称してキャリアと呼び、区別する。

6-6. 職種

ETSS-JMAABでは、自動車制御システム開発のある工程を担当する実務を切り出し、その実務を担当する職業として職務を定義する。ただし通常一人の人間が複数の職務を担当することが多い。

6-7. 責任

ETSS-JMAABでは、キャリア基準で、自動車制御システム開発に関する職種が担うべき責任を各々ごとに明示している。具体的な活動領域を提示することは、職種の活動領域を限定してしまう弊害があるため、あえて責任の範囲を明示し、柔軟かつ現実的な職種が担うべき職掌を表現することとした。

キャリア基準では、キャリアレベル評価を、職種の責任範囲に対するビジネスやプロフェッショナルの貢献の度合いとしている。

このように、キャリア基準は、職種の職掌やキャリアレベルの評価観点として、職種ごとに定義された「責任」を中心としている。先行するITスキル標準を踏まえた形で3段階のキャリアレベルを設定している。

7. 教育プログラム

7-1. 教育プログラムフレームワーク

人材育成を目的とした教育プログラムの構造や仕組みを、教育プログラムフレームワークと定義する。

- 人材育成のための教育プログラムの「入口」「出口」をスキル基準やキャリア基準を用いて定量的に可視化する。
- 定量化された定義を埋める差異にたどり着くための教育科目を適切な履修順序で教育を構成する。

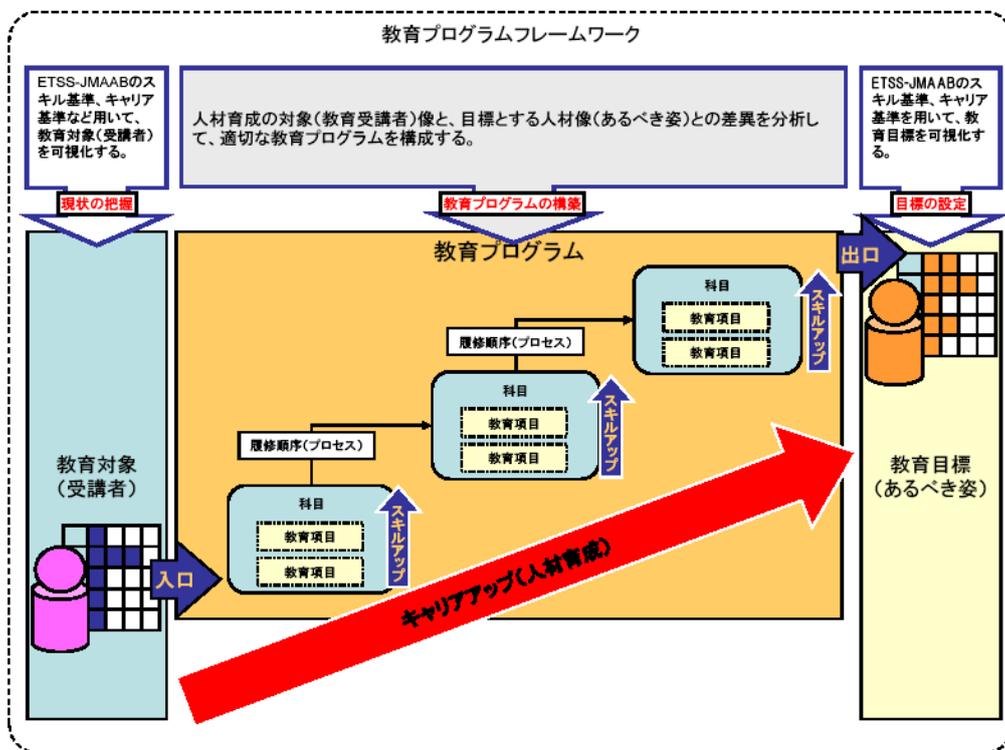


図 16. 教育研修基準 教育プログラムフレームワーク構成要素

7-1-1. 教育プログラム

育成の対象となる人材（受講者）を、教育目標とする人材像（あるべき姿）へ育成を実現するために必要となる教育の体系である。教育プログラムは、1つ以上の科目で構成され、それぞれの科目を効率的に履修できるようガイドする。

教育対象となる人材のスキルや知識の習得状況と、教育目標との差異を明確にし、目指すスキルや知識の領域やレベルに合わせて、適切に科目を選択しまたは組み合わせることにより構成をカスタマイズして個々人に適したプログラムを作成し、運用する。

7-1-2. 科目

特定の技術分野に対する知識やスキルを習得するために、必要となる教育項目の組み合わせ。

科目は、一つのスキルに限定せず、複数のスキル習得を同時に行うこともできる。

科目には、シラバスを用いて、どのようなスキルが習得できるのか、目的が解るようにしなければならない。

シラバスには、以下の項目を記載する。

| 項目 | 概要 |
|-----------|--|
| 教育プログラム名称 | 教育プログラム名称 |
| No | 教育プログラム内で区別する為の番号 |
| 科目名称 | 科目の名称 概要、構成なども記載する |
| 標準時間 | 想定している標準時間 |
| 教育対象レベル | 対象者が持つべき事前の予備知識、標準スキル、想定しているエンジニアの職種、レベルの記載を行います。 |
| 教育目標レベル | この科目の習得によって、レベルアップする対象を示します。 |
| 教育項目 | 科目に含まれるスキル項目、レベルスキル、復讐スキルなどを記載します。 ・教材として使用するテキストの名称 ・機材 |
| 実施形態 | 実施形態です。 OJT か、OffJT か。 指導者の人数、指導者の想定レベル。 講座として提供するのか、WEB トレーニング、ディスカッション 受け入れ人数の規模 |
| 育成成果の確認 | 効果の評価方法を記載します。 |

実際の運用時には、上記以外に、日程、費用など開催に必要な情報も記載が必要です。

8. M B D用語

(1) E C U (Electronic Control Unit)

電子制御ユニットの略語

(2) E T S S (Embedded Technology Skill Standards)

独立行政法人情報処理推進機構が策定した組込みS/W開発力強化のための体系的な基準

(3) E T S S - J M A A B (J M A A B M B D Technology Skill Standards on E T S S)

JMAABがETSSに準拠して策定した、自動車分野のMBDを対象としたスキルの体系的基準で、スキル標準、スキル基準とキャリア基準から構成される

(4) H I L S (Hardware In the Loop Simulation)

パワートレインやシャシーなどの制御対象挙動をリアルタイムの仮想環境で実現し、実ECUや実部品を接続して評価する装置

(5) J M A A B (Japan M B D Automotive Advisory Board)

国内の自動車メーカーとECUサプライヤのMATLABユーザ会

(6) M B D (Model Based Development)

モデルベース開発の略語

(7) M I L S (Model In the Loop Simulation)

制御機能の全体やロジックの一部をモデルで記述し、仮想環境で評価する装置

(8) P I L S (Processor In the Loop Simulation)

自動車制御システム開発で対象とするシステム構成要素として、制御装置の演算マイコン等の挙動について仮想環境で実現し評価する装置

(9) S I L S (Software In the Loop Simulation)

自動車制御システム開発で対象とするシステム構成要素として、制御アプリケーションの全体もしくは一部をS/Wで記述し、仮想環境で評価する装置

(10) キャリブレーション

燃費・排ガス規制などのシステム機能への要求と乗員の利便・快適性などの感性に関する要求を満足させるためにシステム制御パラメータを全体最適化すること

(11) Simulink API

A P I はアプリケーション プログラミング インターフェースの略で、Simulink モデルの様々な操作や設定などを、MATLAB のコマンド ラインまたはスクリプトを利用して実行できるようにした関数インターフェースのこと。

(12) 仕様ベーステスト

ブラックボックステスト。コンポーネントやシステムの内部構造を参照することなく、それらの機能的あるいは非機能的な仕様の分析に基づいてテストケースを作成して行き、“入力”に対する“出力”が正しいか否かを検証する。同値分析、境界値分析、デシジョンテーブル・テスト、状態遷移テスト、ユースケース・テストなどの手法がある。外部仕様に基づいてテストを行うので、仕様に表れない内部的・潜在的な不具合についての検証ができない。また、何らかの欠陥が重なって“たまたま”結果が仕様どおりだった場合についても欠陥を検出できない。

(13) 経験ベーステスト

テスト設計の力量に影響されるが、仕様ベースや構造ベースのテスト技法では見つからないような欠陥も見つけられることがあるため、仕様ベースや構造ベースのテスト技法と併用すると効果的である。ソフトウェア開発でありがちな誤り（エラー）を、テスト担当者が推測してテストケースを作成するエラー推測や、過去不具合などから抽出した観点から行う再発防止などがある。

(14) 構造ベーステスト

ホワイトボックステスト。コンポーネントやシステムの内部構造の分析に基づいて網羅的にテストケースを作成して行うテストで主に単体テストに用いられる。漏れなくテストを行うという目的の実現度を計測するために、カバレッジ（網羅率：テスト対象となった割合を示すもの）を指標として用いる。仕様にはない実装をチェックしたり、単純なコーディングミスなどを検出できるが、どの制御パスが全て矛盾なく動作していても、それが仕様に合致しているかどうかは確認できない。

(15) M C / D C

Modified Condition/Decision Coverage の略。

MC/DCは、国際技術標準 DO-178B (RTCA) に準拠している。これは、S/W証明のための基準を指定したものであり、対象となるS/Wには、航空業界で使用される重要機器及びシステムのためのリアルタイム組込みシステムも含まれる。DO-178Bに従うと、完全な(100%) MC/DCを得るには、次の3つの条件を満たす必要がある。

- ・各「判断文」が、少なくとも1回全ての可能な結果を得ている。
- ・1つの「判断文」中の各条件が、少なくとも1回全ての可能な結果を得ている。
- ・1つの「判断文」中の個々の条件が、単独で全体の「判断文」の結果を左右する

VI. 図表一覧

| | |
|---|----|
| 図 1. 制御システムにおけるモデルの位置づけ | 11 |
| 図 2. システム開発プロセス | 12 |
| 図 3. MBDプロセス | 13 |
| 図 4. S/W開発プロセス | 13 |
| 図 5. システムと制御システムの関連 | 14 |
| 図 6. ETSS-JMAABの構成 | 23 |
| 図 7. 開発者のポテンシャルを複数のスキルレベルで可視化する | 24 |
| 図 8. 職種や専門分野が担う責任や役割の達成度合いをキャリアレベルとして可視化 .. | 25 |
| 図 9. スキルと知識の関係 | 29 |
| 図 10. ETSS-JMAABスキルフレームワーク | 30 |
| 図 11. 職種とキャリアレベル | 39 |
| 図 12. X社A部門MBDエンジニアスキル標準の位置づけ | 52 |
| 図 13. X社A部門MBDエンジニアスキル標準の作成 | 53 |
| 図 14. スキル(技能)と技術力 | 61 |
| 図 15. 技術者と技能者 | 61 |
| 図 16. 教育研修基準 教育プログラムフレームワーク構成要素 | 63 |
| | |
| 表 1. 自動車に関する知識カテゴリ | 29 |
| 表 2. 技術要素スキルカテゴリ (CVT制御開発の例) | 33 |
| 表 3. キャリアレベルの定義 | 38 |
| 表 4. キャリアレベルとプロフェッショナル評価の達成指標 | 39 |
| 表 5. 職種と責任及び役割の対応表 | 40 |
| 表 6. 職種とプロセスの関係 (参考例) | 43 |
| 表 7. 開発環境エンジニアのための Simulink API スキル定義 | 50 |