

クラウド連携組み込みシステムの評価 —LLM (大規模言語モデル) の活用—

細川 守*1, 石田 倫章*1, 松尾谷 徹*2, 古畑 慶次*3

*1 株式会社デンソー, 〒448-8661 愛知県刈谷市昭和町1丁目1

E-mail: mamoru.hosokawa.j5n@jp.denso.com, michiaki.ishida.j3w@jp.denso.com

*2 有限会社デバッグ工学研究所, 〒206-0022 東京都多摩市聖ヶ丘4丁目16番地1

E-mail: matsudani@gmail.com

*3 有限会社生産経営研究所, 〒448-0857 愛知県刈谷市大手町1丁目32番地

E-mail: xddp@nifty.com

概要 組み込みソフトウェアは、製品のドライバーとしてROMに固定された時代から、サービスとしてリリース後もアップデートされる時代になってきている。また、システム開発においても、レガシー開発時代の派生開発と、現代のモダン開発、さらにLLM(大規模言語システム)のアシストによるポストモダン開発へと急変している。我々は、この変化に対応できる技術リーダー育成を20年間継続し、育成プログラムを進化させてきた。その経緯と24年度のアプローチを紹介する。「クラウド連携組み込みシステムの評価」は、クラウドと組み込みのハイブリッド構成のシステムテスト設計にLLMを導入した実践的な研修成果を示す。

キーワード LLM, 技術リーダー, クラウド連携, システムテスト

Evaluation of cloud-linked in-vehicle embedded systems —Utilizing LLM (Large Scale Language Models)—

Mamoru Hosokawa*1, Michiaki Ishida*1, Toru Matsudani*2, Keiji Kobata*3

*1 DENSO CORPORATION, 1-1 Showa-cho, Kariya City, Aichi Prefecture, 448-8661

E-mail: mamoru.hosokawa.j5n@jp.denso.com, michiaki.ishida.j3w@jp.denso.com

*2 Debugging Engineering Research Institute, Ltd., 1-16-1 Seigaoka, Tama City, Tokyo 206-0022

E-mail: matsudani@gmail.com

*3 Production Management Research Institute, Ltd., 1-32 Otemachi, Kariya City, Aichi Prefecture 448-0857

E-mail: xddp@nifty.com

Abstract Embedded software has moved from the era when it was fixed in ROM as a product driver to an era when it is updated as a service even after release. System development has also undergone a rapid transformation from the derivative development of the legacy development era to modern development in the present day, and finally to postmodern development assisted by LLMs (large scale language systems). For the past 20 years, we have been training technical leaders who can respond to these changes and have evolved our training programs. We will introduce the background and our approach for FY2012. We will present "Evaluation of cloud-linked in-vehicle embedded systems" as a practical training outcome that introduces the LLM to system test design for a hybrid configuration of cloud and embedded systems.

Keyword LLM, Technology Leader, Cloud Integration, System Testing

1. はじめに

車載組み込みソフトウェア開発は、車両がSDV (Software Defined Vehicle) へと進化することで、大きな変化に直面している。従来のハードリアルタイム制御技術やプロセスでは対応が難しくなり、中堅技術者の過去の経験が役立たない状

況が生じている[1,2]。

市場の変化は、中堅技術者に新たな環境下で、未経験の技術に対しても「技術リーダー」としての役割を求める。しかも、納期や品質の要求は変わらず、激務の中でのキャリアチェンジだ。技術者にとって、新たな技術を獲得するのは日常的な活動だが、組み込み開発は、変化を避け安全性を

確保する派生開発であり、大きなギャップが生じる。

このギャップを埋め、キャリアチェンジを速く成功させるには、支援が必要だ。我々は、この課題に 20 年前から「技術リーダー育成研修」に取り組み実践してきた。近年の大きな市場変化に対応できるプログラムを進めており、その事例を含めて報告する[3]。

本報告では、技術リーダー育成の取り組みと、新たに導入した LLM（大規模言語モデル）を活用した GAI（生成 AI）の事例として、クラウド連携組み込みシステムの評価におけるチャレンジを紹介する[4,5]。

2 章では、組み込みソフトウェア開発におけるスキルチェンジの背景と課題を示し、3 章では対策として実施している研修について詳しく説明する。4 章では、研修の主プログラムである「実務課題に対する論文研修」の例として、クラウド連携組み込みシステムの評価について説明する。5 章では、研修事例と研修の効果について考察する。

2. 背景

技術リーダーに求められるスキルチェンジの課題背景と、従来型の研修や OJT の問題点について説明する。

自動車産業は電気自動車や自動運転などの新技術によって大きく変化した。この変化は、SDV（Software Defined Vehicle）と呼ばれ、コア技術がエンジン制御や機構制御から、ソフトウェアによる機能やサービスへとシフトしたことを示している[1,2]。

SDV の「ソフトウェア」と組み込み開発の「ソフトウェア」は同じ用語を用いているが、まったく異なる技術分野だ。そのため、ソフトウェアを支える技術者のスキルや仕事の進め方も大きく異なる。

企業経営の観点から見ると、人的資源は準終身雇用のもとで、ジョブが変われば役割も変わる仕組みとなっている。ソフトウェア技術者に関しても同様であり、SDV トрендへの対応策として技術者の部署異動が行われている。

技術者のスキルチェンジは、従来の OJT 方式では、技術者の努力と業務を通じて解決されることが一般的だ。しかし、SDV と組み込み開発の技術ギャップは大きく、これまでの OJT 方式では対応が難しく、開発業務にも悪影響を及ぼす恐れがある。

OJT 方式は、新人レベルのスキル獲得やスキルチェンジには有効だが、高度なスキル習得を短期間で達成するには適していない。そのため、SDV へのビジネスシフトにおける大きな課題の一つとして、ソフトウェア技術リーダーの適切なスキルチェンジが挙げられる。

表 1: 開発スタイルの概要比較。

スタイル	レガシー	モダン	ポストモダン
例	派生開発	アジャイル	DevOps
リリース先	ROM	顧客向け AP	サービス
価値観	安全、確実	顧客価値	市場価値
プロセス	規範準拠	フィードバック	連続した拡張
スキルチェンジ	深化	AP 対応	先端技術
協働範囲	狭い	中	拡大
例	メカ、エレキ	顧客、インフラ	巨大なシステム

表 1 は、開発スタイルの概要を比較したものだ。レガシーとは、派生開発などの組み込み開発スタイルを指す。モダンとは、オブジェクト指向やアジャイル開発など、21 世紀初頭に確立したソフトウェア工学のスタイルを指す。ポストモダンは、ネットを介して新たなサービスを継続的に提供する開発スタイルだ。

SDV への移行は、レガシースタイルとポストモダンスタイルの共存と協働が生じ、技術だけでなく、開発文化やマネジメントスタイルに大きなギャップが生じる。

このギャップを乗り越えるには、何らかのサポートとしての研修が必要だ。しかし、知識の学習だけでは非効率であり、より実践的な研修方式が求められる。

3. 研修方式

この章では、背景で述べた課題への対策として実施している「技術リーダー研修」について説明し、加速する市場の変化に対応するための研修プログラムのチャレンジを示す。

デンソーでは技術者育成の仕組みとして、業務遂行に必要な固有技術やノウハウを段階的に習得するテクニカルスキル研修と、技術開発や事業展開を牽引する技術リーダー研修を実施している。技術リーダー研修は、生産技術、パワーエレクトロニクス、ソフトウェア工学など、各技術分野ごとにコースを構築しており、ソフトウェア工学コースでは、「ソフトウェア開発における問題解決を牽引できる技術者の育成」を目標に、2004年度から入社 5～10 年程度の中堅社員を対象に実施している。

技術の変化に対応し、研修スタイルは 1 期から現在の 5 期へと進化を続けている。

・第 1 期 (2004～) : 技術リーダーのための体系的工学習得

ソフトウェア開発現場には多くの課題が山積していた。技術者は過去の経験に基づく開発を続けており、ソフトウェア工学の知識不足が課題解決を阻害していたことから、ソフトウェア工学を実践できる技術者の育成を目指した。研修では、ソフトウェア工学の講義や演習を行い、習得した知識を各自の課題に適用し、職場での展開方法を発表した。

・第 2 期 (2008～) : 課題論文研修の開始

研修生の問題解決経験が不足しており、職場で得た知識を活用できない問題があった。そこで、開発の問題分析を実施し、課題形成・解決を論文にまとめる研修科目を追加した。

・第 3 期 (2012～) : リーダー規範の確立

技術知識だけでなく、リーダーとしての姿勢や行動を身につけるカリキュラムを導入した。職場の多忙な環境下で、技術リーダーとしてメンバーを鼓舞し、難題に挑戦する意欲を育むためのリーダーシップ教育を強化した。

・第 4 期 (2020～) : 課題のチーム・組織単位への拡大

研修生個人の課題が局所化し、過去の研修成果が十分に横展開されていない問題があった。課題を抽象化し、共通課題としてチームや組織に拡大す

ることで、知識の共有と活用を促した。

・第 5 期 (2024～) : 技術チャレンジの追加
SDV や LLM といった新技術への対応が求められるようになった。組み込みソフトウェアの枠を超え、ECU 統合や LLM 技術への対応が必要になったため、研修内容に新技術へのチャレンジを加えた。

研修方式の変化は、ソフトウェア工学を実務で活用するために必要なスキルである「問題解決」「リーダーシップ」に加え、新技術への「チャレンジ」を組み込む形で進化している。今期は LLM の活用をテーマに、派生開発と SDV 開発の両分野で研修を実施している。次章では、SDV 開発分野の課題研究について紹介する。

課題研究は 3 つのステージで構成され、それぞれ約 3 カ月の期間で進められる。新技術のサーベイを経て、各研修生のテーマ設定へと進み、最終的には職場への新技術導入に挑戦する。挑戦結果は論文にまとめ、役員に報告するとともに組織の共通ナレッジとして資産化している。テーマ設定から職場への技術導入においては、社内のアドバイザーによる支援が欠かせない。設計課長、シニアエンジニア、研修修了生が社内のアドバイザーを務め、研修生に対して 1 対 1 で支援を行っている。次章では、クラウドと車載システムを連携した開発へのチャレンジ事例を取り上げる。

4. 実践例

4.1. クラウド連携組み込みシステムの全体像

クラウド連携組み込みシステムは、図 1 の通り、車両とセンタの両方が存在し、それらが繋がるシステムであり、ここでは車両のカメラデータ収集システムを対象とする。車両のカメラデータ収集システムは、例えば車速が 80km/h 以上となった時に、車両前方のカメラデータをセンタにアップロードするシステムである。

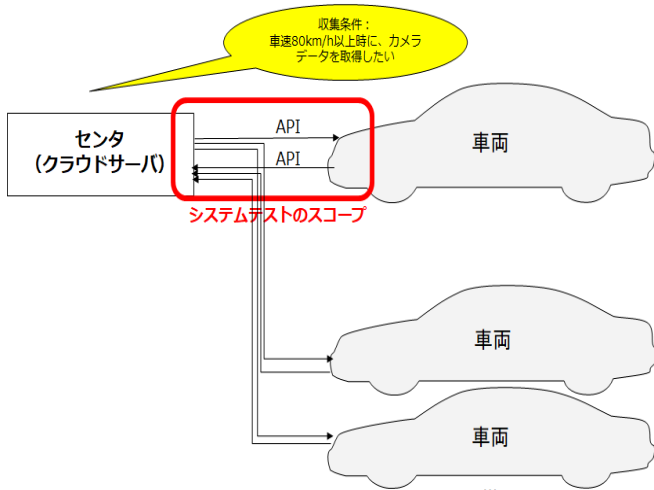


図 1: システムスコープ。

図 1 の通り、車両ーセンター間は API 通信が行われ、クラウド連携組み込みシステムのシステム評価では、本質的にはその API をテストする形となる。システム評価を行う上では、多数のテストケースを作成する必要があるが、テストケース作成時に大きく 2 つの課題があり、簡単にテストケースを作成することができない。

課題 1. ある機能に着目しテストケースを作成する際に、その機能に関わる仕様がクラウドと車両に分散している。

課題 2. システム要求仕様と、クラウドと車両の設計・実装 (版数毎リリース) が一致しておらず、ギャップがある。

テストケース作成を行う上では、図 2 のようにシステム要求仕様書のみならず、クラウドも含めた各製品の設計仕様の把握と関係付けが必要である。一方、我々のような車載システムの技術者は組み込み分野のシステム開発のスキルは有しているものの、クラウド側の開発スキルについては未経験で未知の分野である。

特定機能のテストケースに関係する仕様を洗い出して関係付けが必要

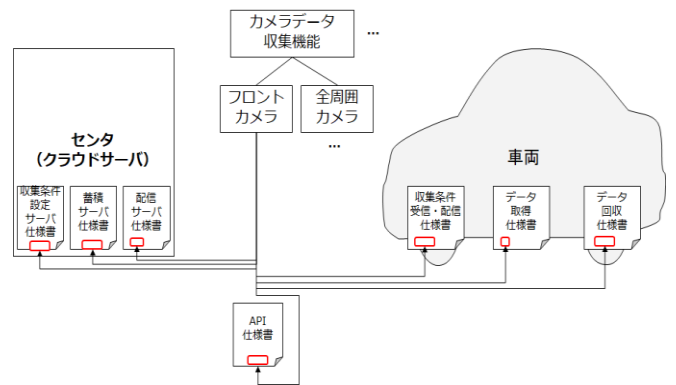


図 2: テストケース作成時の課題。

そこで、これらの課題解決を行うため、研修の一環として、新技術 (LLM) を用いて、クラウド連携組み込みシステムの評価のテストケース作成にチャレンジした。そこで本稿では、図 1 の赤枠部分にスコープを限定し、クラウド系の API の仕様や各種要求仕様、設計仕様を LLM に理解させた上で、LLM でテストケース作成を行えるかを模索する。

4.2. テストケースの抽出

LLM を用いたテストケース作成にチャレンジするにあたり、まずはシンプルな事例でトライする。ここでは、API 仕様書に定義されている範囲の値を組み合わせ、車両のカメラデータやメタデータ (CAN データ) を取得するという最もシンプルな正常系のテストケースを想定する。車両のカメラデータ収集システムでは、センターと車両では図 3 のやりとりが発生する。その中で本稿では、車両からの収集条件のリクエストをセンターが受けとり、センターから車両に収集条件をレスポンスするところに絞り、テストケースを検討する。

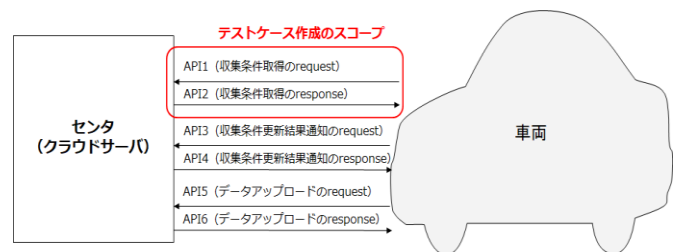


図 3: テストケース作成のスコープ。

各テストケースには下記の構造をもたせ、テストケース作成の際には下記の内容を記載することとした。

- テストケースで評価する機能
- テストケースで API1, API2 に入力するパラメータ (API1:31 個, API2:105 個)

下記の STEP1~STEP4 のような流れでテストケース作成を進めた。上記の API1 および API2 のパラメータの定義については API 仕様書に、こういう機能があるシステムなのかはシステム要求仕様書に記載されている為、それらを入力としてテストケースを作成した。ただ、それだけではどのようなテストケースを作成すれば良いのか LLM/GAI が判断できなかったため、テストケースの構造 (定義) も入力し、LLM/GAI でテストケースを作成した。

STEP1 は今回のテストケース作成の 4 つの STEP の中で最も苦労した STEP である。API1 と API2 のフィールドを正確に抽出することが困難であり、その理由として、今回の API 仕様書が図 6 の通り、API1~6 の全 API 仕様シート毎に分けられており、1つのシートの中にリクエストとレスポンスが記載されていて情報量の多い Excel 表となっていることが挙げられる。更に、各 API のフィールドも大分類、中分類で列が別れていて、フィールド抽出が難しい API 仕様となっていたことも挙げられる。このような仕様書では多くの場合、共通的な仕様フォーマットが存在せず、製品や職場固有のフォーマットとなっており、LLM/GAI がそれらのフォーマットを正確に識別できるようチューニングが必要となる。

- STEP1. LLM/GAI に API 仕様を入力
- STEP2. LLM/GAI にテストケースの構造を入力
- STEP3. LLM/GAI にシステム要求仕様書を入力
- STEP4. LLM/GAI に作成したいテストケースを指定して依頼し、テストケースを作成

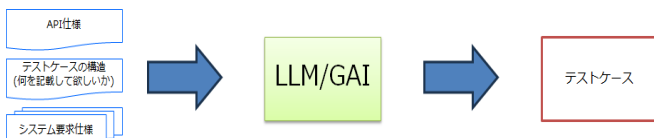


図 4: LLM/GAI を用いたテストケース作成。

STEP1 : API 仕様の入力

STEP1 のイメージを図 5 に示す。

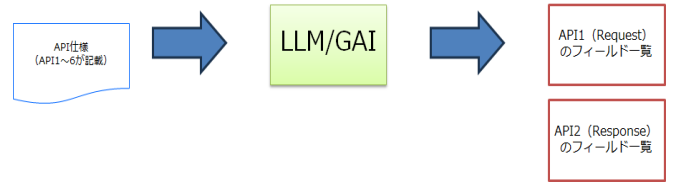


図 5: API 仕様の入力

API仕様書

インターフェースID : CMR010
インターフェース名 : 収集条件取得

リクエスト

フィールド1 (フィールド名、値の定義など)	
大分類	フィールド2 (フィールド名、値の定義など)
	フィールド3 (フィールド名、値の定義など)
中分類	フィールド4 (フィールド名、値の定義など)
	フィールド5 (フィールド名、値の定義など)

レスポンス

フィールド6 (フィールド名、値の定義など)	
大分類	フィールド7 (フィールド名、値の定義など)
	フィールド8 (フィールド名、値の定義など)
中分類	フィールド9 (フィールド名、値の定義など)
	フィールド10 (フィールド名、値の定義など)

①全体像シート ②API1,2シート ③API3,4シート ④API5,6シート ⑤共通ヘッダシート

図 6: API 仕様書のイメージ。

STEP2 : テストケースの構造の入力

STEP2 のイメージを図 7 に示す。

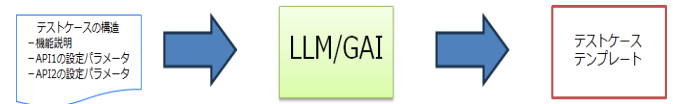


図 7: テストケース構造の入力。

STEP3 : システム要求仕様書の入力

STEP3 のイメージを図 8 に示す。



図 8: システム要求仕様書の入力。

STEP4 : テストケース作成

STEP4 のイメージを図 9 に示す。

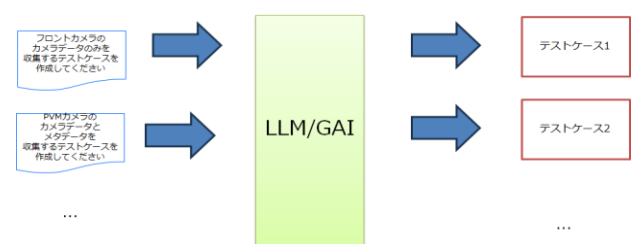


図 9: テストケース作成。

本稿では、カメラデータのアップロード機能のテストケース作成にトライした。STEP1~3 の入力を行った上で、LLM/GAI に STEP4 でテストケース作成依頼を行った。その際、テストケース作成の前に、参照する API 仕様書を絞ったり、テストケースの構造を明確化するという工夫を行ったことで、依頼と合致した API パラメータにてテストケース作成結果が得られ、期待したテストケースを作成することができた。その結果、テストケース作成における LLM の活路を見出した。

【テストケース1 (フロントカメラデータのみ収集) の API パラメータ例】

```
upload_type=UT_CAMERA_IMAGE_DATA_ONLY  
image_data_type=CT_FRONT_CAMERA
```

【テストケース2 (PVM カメラとメタデータを収集) の API パラメータ例】

```
upload_type=UT_BOTH  
image_data_type=CT_SURROUND_CAMERA
```

5. 評価と考察

我々が行っている、中堅技術者に対する技術リーダー育成の観点から評価し、効果について考察する。第一の評価は、事例として示した「クラウド連携組み込みシステムの評価」について示し、その後、研修として「新技術チャレンジ」の評価について述べる。

5.1. クラウド連携組み込みシステムの評価

「クラウド連携組み込みシステムの評価」は、研修生の職場における実務に近い分野への技術展開を行った。システム連携は、異なる開発文化と開発技術、職場固有の仕様書体系や文書体系を持っており、ハイブリッド開発であった。

このようなハイブリッド開発環境において、システムの連携した機能や非機能の評価することの難しさに遭遇した。テスト設計の入力となる、様々な仕様の所在確認、仕様フォーマットの正確な識別、技術用語のシノニムなどで手間取った。この段階では、全文検索など既存のツールと LLM との組み合わせが有効と思われる。

テストケース抽出段階では、個々の仕様書間で記述の揺れがあり、導入段階において、プロンプトの作成と評価/修正が必要であった。プログラムのデバッグ同様、何が正しいかを判断できるスキルが無いと、導入が困難であった。

システム開発における仕様書は多くの場合、共

通的な仕様フォーマットが存在せず、製品や職場固有のフォーマットとなっている。そのため、LLM 活用を考えた際に、LLM/GAI がそれらの固有フォーマットを正確に識別できるようなチューニングが課題となる。一方、その課題をクリアすることで多数の API パラメータを含むテストケースを瞬時に作成することができ、テストケース作成における LLM の活路を見出した。

5.2. チャレンジ研修の評価

今期の研修で試みた「新技術へのチャレンジ」は、5つの分野で展開した。派生開発などレガシー系3件、SDV系2件であった。

研修生の職務に近い技術活動を、新技術活用の実践演習の対象に選ぶ場合、研修の効果を左右する一番の要因は、研修生がそのテーマに対する興味の大きさである。

従来は、職場の状況から、技術やツールを選択するアプローチであった。良い技術が見つからなかったり、効果が低かったりするケースも生じ、実戦形式研修の課題でもあった。

今期は、汎用性の高い新技術 LLM/GAI により、5分野すべてで、研修生は技術に興味を持ち、例年よりモチベーションが高いと思われる。客観的な比較方法を検討中だが、最終レポートである論文の充実程度と量から推測できる。

6. おわりに

自動車の SDV 進化に伴い、車載組み込みソフトウェア開発に従事する技術者にはスキルチェンジが求められているが、OJT だけの対応は困難である。どのような訓練や教育が必要なのか？この問いに答え続けてきた「技術リーダー研修」の20年間の経緯を紹介した。

工学知識、問題解決、リーダーシップに加え、現在は LLM や GAI などの新技術への「チャレンジ」を取り入れようとしている。第4章では、チャレンジ研修の一端として「クラウド連携組み込みシステムの評価」を紹介した。

研修生が実務で課題研究にチャレンジできるのは、研修生の人数を上回る社内のアドバイザーと呼ばれる諸先輩方の支援のおかげである。人材育成を支える企業文化に感謝します。

文 献

- [1] Sonko S, Daudu CD, Osasona F, Monebi AM, Etukudoh EA, Atadoga A. The evolution of embedded systems in automotive industry: A global review. World Journal of Advanced Research and Reviews 2024;21:096-104. <https://doi.org/10.30574/wjarr.2024.21.2.0420>.

[2] Blanco DF, Le Mouël F, Lin T, Escudié M-P. A comprehensive survey on Software as a Service (SaaS) transformation for the automotive systems. IEEE Access 2023.

[3] 古畑慶次, 上杉卓司, 足立久美, 青山幹雄, others. 高度ソフトウェア専門技術者育成のためのカリキュラム開発方法論の提案と評価. ソフトウェアエンジニアリングシンポジウム 2014 論文集 2014;2014:15-20.

[4] Fan A, Gokkaya B, Harman M, Lyubarskiy M, Sengupta S, Yoo S, Zhang JM. Large Language Models for Software Engineering: Survey and Open Problems. 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), 2023, p. 31–53. <https://doi.org/10.1109/ICSE-FoSE59343.2023.00008>.

[5] Wang S, Yu Y, Feldt R, Parthasarathy D. Automating a Complete Software Test Process Using LLMs: An Automotive Case Study 2025.