

# 組込みシステムにおける 検証アーキテクトと育成プログラム

西原 秀明<sup>※1</sup>大野 喜宏<sup>※2</sup>木村 浩司<sup>※3</sup>瀬野 恭彦<sup>※4</sup>

組込みシステムの検証における高度人材育成についての活動を紹介する。システム開発全体の中で検証を位置付け、幅広い視野とアイデアを基に検証をデザインし実施する検証アーキテクトの育成を目的に、その役割を定義し、必要なスキルを習得するためのカリキュラムを策定した。「組込み適塾」にてカリキュラムを試行し、有効性を評価した。

## Fostering Verification Architects for Embedded Systems

Hideaki Nishihara<sup>※1</sup>, Yoshihiro Oono<sup>※2</sup>, Koji Kimura<sup>※3</sup>, and Yasuhiko Seno<sup>※4</sup>

This article introduces activity for fostering verification architects in embedded systems development. Verification architects specify verification activities in overall system development, conduct verifications with great insight, and contribute to the quality of products. A curriculum for verification architects is considered based on their expected roles in each phase of the system development process. The result was applied in Kumikomi-Tekijuku, an education program for embedded system architects.

### 1 導入

#### 1.1 組込みシステムと検証

生活を支える機器やシステムにおいて、ソフトウェアが制御や機能の中心的役割を担うようになり、高機能化、複雑化が進んでいる。それに併せて機器やシステムの品質・信頼性の確保に対する考え方も変化し、様々な取り組みがなされている。

組込みシステムの開発では多くの場合、テスト工程に大きな比重をおいてシステムの検証を行っている。しかし例えば際どい操作に対するテストや妥当性確認では、観点の適切な選択や柔軟な発想といった属人的な作業が結果に影響することもあり、テスト技術の水準を維持し

ていくのは容易ではない。またテスト工程は開発プロセスの後半に位置し、実施に制約がかかりがちで開発全体へのフィードバックも十分に行えない。検証工程を改善する余地はまだ大きく残っており、とくに工学的な観点をとることで開発全体を改善し信頼性の確保に貢献すると考えられる。

#### 1.2 組込み適塾とキャリアガイド

組込みシステム産業振興機構(以下振興機構と呼ぶ)<sup>[1]</sup>では、組込みソフトウェアに重点をおいた高度人材育成プログラム「組込み適塾」を2008年より実施している。システムアーキテクト、すなわち開発対象である組込みシステムのアーキテクチャを把握し技術視点から開発を統

※1 国立研究開発法人 産業技術総合研究所

※2 株式会社インサイト

※3 AVC テクノロジー株式会社

※4 組込みシステム産業振興機構

括する技術者の育成を軸とし、開発の段階に合わせた「実装エンジニアリングコース」「アーキテクチャ設計コース」「アドバンストコース」の3つのコースを設定している。組込みシステム開発企業がカリキュラム検討に多く参画し、実践的な知識を体系的に身に付けることができる特徴的なプログラムである。

振興機構では、組込み技術者のキャリアガイド並びにキャリアマップ<sup>[2]</sup>を作成し、組込み適塾カリキュラムとの対応付けを行った。図1に2014年版のキャリアマップを示す。「実装エンジニアリングコース」がソフトウェアエンジニア、ハードウェアエンジニアのキャリアを広くカバーしており(図中(C)で示されている部分)、同様に「アーキテクチャ設計コース」がミドルレベルのシステムアーキ

テクトを中心にプロジェクトマネージャとソフトウェアエンジニア、ハードウェアエンジニアを(図中(B))、「アドバンストコース」がハイレベルからミドルレベルのシステムアーキテクトをカバーしている(図中(A))。

キャリアガイドにはソフトウェアエンジニアやシステムアーキテクト、プロジェクトマネージャを含む「開発系キャリア」、開発プロセス改善スペシャリストや開発環境エンジニアを含む「支援系キャリア」に加え、検証・テスト担当者のキャリアとして「検証系キャリア」が設定されており(図1の点線枠)、組込みシステム開発におけるテストの重要性を示している。一方で従来の組込み適塾のカリキュラムでは設計視点に重点がおかれていたため、キャリアマップを十分に網羅していない状態にあった。

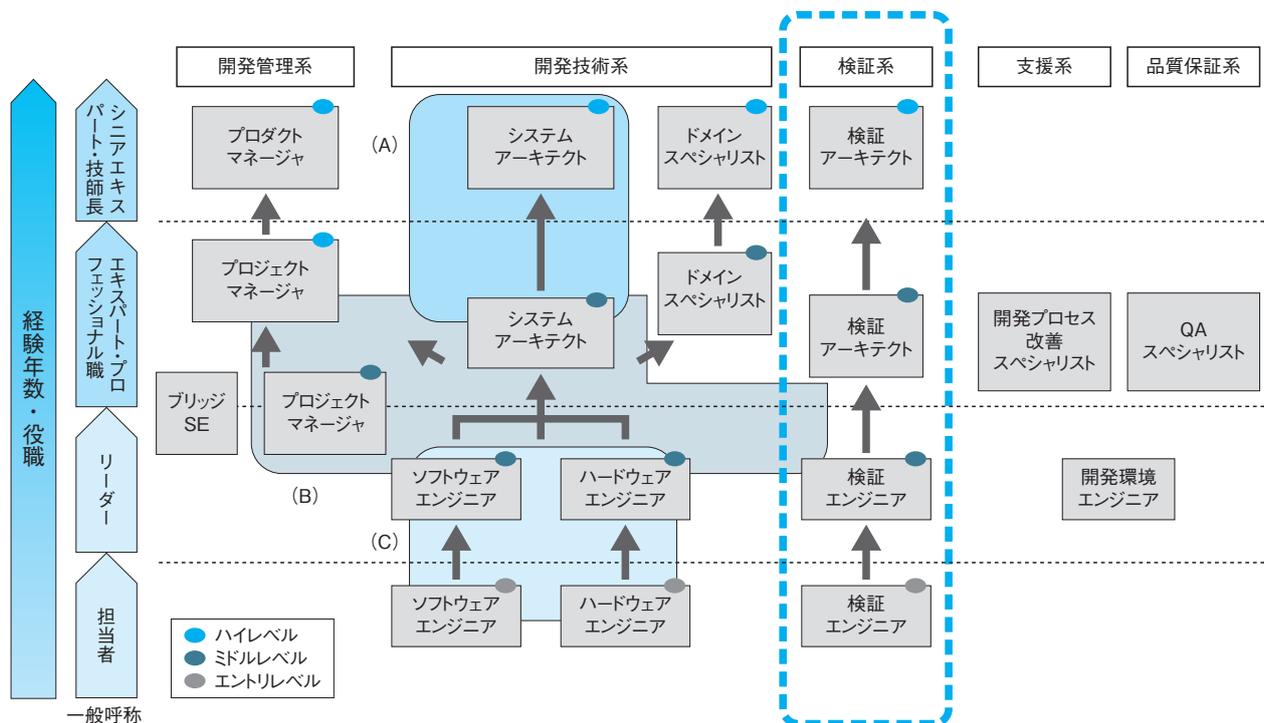


図1 キャリアマップとカリキュラムの網羅性

### 1.3 検証系人材の育成に向けて

このような状況のもと振興機構関連の企業から検証に関する教育への大きな要望があり、組込み適塾にて検証系の内容を拡充することとなった。大野喜宏主査の下、検証系人材育成サブワーキング(以下サブワーキングと呼ぶ)が組織され、本論文の著者を含め10名で活動を開始した。サブワーキングのメンバーは組込みシステム開発企業で開発経験を積んでおり、現場における検証の実態や課題を熟知している。またメンバーの多くが管理者的な立場にあり、開発全体の視点で必要とされる知識や人材について議論を行った。

サブワーキングは2014年度に6回開催され、現状と課題の共有を念頭においた討論、育成すべき人材像の明確化に続いてカリキュラム策定へと議論を進めた。サブワー

キングでは、将来必要とされる検証技術者は開発全体における検証工程を位置付け、検証の各作業を設計するアーキテクトであるとの認識に至り、テスト観点の抽出、テスト結果の適切な分析とフィードバック、再発防止のための活動・提案をテーマとしてカリキュラムを検討した。

2015年には組込み適塾「検証アーキテクティング科目」として策定したカリキュラムが試行された。サブワーキングでは2015年度に6回の会合を行ってカリキュラムを評価し、改善点を整理した。その結果は2016年に実施されるカリキュラムに反映される。

### 1.4 本稿の内容

以上のように、著者らはサブワーキング活動の中で、組込みシステム開発における検証技術の向上を目指してカリ

キュラムを策定し実施した。検証視点のアーキテクトという概念は当時一般的ではなく、その役割やスキルについて一から検討して整理し、適切なカリキュラムを策定する必要があった。本稿ではその詳細と結果について述べる。

まずサブワーキングでは組込みシステム開発における検証アーキテクトの人材像について議論し、軸となるスキルと開発プロセスの中での位置付けを整理した。その結果を第二節で説明する。

育成したい人材像に基づいて、テーマ設定、順序、分量などを考慮しカリキュラムが作られる。サブワーキングの議論では、テスト観点の抽出、テスト結果の適切な分析とフィードバック、再発防止のための活動・提案が検証アーキテクト育成のテーマとして設定された。そこで、これらの3つのテーマに対応する3つの講義と、検証アーキテクトの活動全体を俯瞰する講義、合わせて4つの講義からなるカリキュラムが定められた。検証アーキテクトを対象とするため、個別の検証技術の知識よりは開発プロセス全体の視点から検証の効果を高める知識に重点がおかれていることが特徴である。カリキュラム策定の方向性や各講義の内容については第三節で説明する。

カリキュラムの試行と評価の結果については第四節で説明する。評価はサブワーキングによる講義内容のレビューと受講者の事後アンケート、講師のコメントを基にして行った。改善すべき点が幾つか指摘されたものの、検証アーキテクトに有益な知識や技能を提供するプログラムであることが確認された。今後、内容の拡充や見直しを進めつつ、継続してカリキュラムを実施し、検証アーキテクトの育成を続けていく計画である。

## 2 検証アーキテクトの人物像

### 2.1 検証アーキテクトについての現況

検証アーキテクト或いはテストアーキテクトはまだ一般的に知られた職種ではないが、テストエンジニア<sup>\*1</sup>の一段上の視点から検証工程を捉え推進する技術者やスキルが必要とされていることは以下のように幾つかの文献で指摘されている。

ソフトウェアテストに関する国際規格IEC/ISO/IEEE29119<sup>[3]</sup>では、組織のテスト方針や開発計画を基に個々のプロジェクトにおけるテストの戦略や明示的な計画を定めるテスト管理プロセスが記述されている。

Microsoftではテストアーキテクトをおき、開発のエンジニアリングプロセスの改善に責任を持つポジションとしている<sup>[4]</sup>。更に同社のテスト体制について書かれた書籍<sup>[5]</sup>の第二章では、上級のテストエンジニアがテストアーキテクトの役割を果たしているケースがあることを指摘している。これらのエンジニアはテストのためのインフラ構築や複雑なテストを創造する際の事項の評価などを担当し、技術的観点から検証工程の向上に貢献する。

JSTQB<sup>[7]</sup>ではソフトウェアテスト技術者のスキルレベルをFoundation/Advanced/Expertの三段階に分けている。

Advanced levelでは個々のテストプロジェクトにおいてパフォーマンスを向上させるテストマネージャ、テクニカルテストアナリスト、テストアナリストの3つの資格が設定され、更にExpert levelではテストプロセスの改善、管理、テストの自動化、セキュリティテストといった項目の資格が設定されている。

そして組込み技術者向けキャリアガイド<sup>\*2 [2]</sup>では、検証アーキテクトを「システムアーキテクトと共にアーキテクチャ観点で品質を確保し保証する技術者」と定義している。検証アーキテクトはシステムアーキテクトの成果のレビューなど上流の工程にも参画することが想定されており、その結果、テスト工程（とくにシステムテスト工程）にかかわるスキルに加えてシステム設計などシステムアーキテクトと同範囲のスキルが必要とされている。

### 2.2 期待される役割

これらの状況を踏まえ、サブワーキングにて組込みシステムの領域における検証アーキテクトの人物像を整理した。図1に示したキャリアマップでは、検証エンジニアの直接のキャリアアップ先としてミドルレベルの検証アーキテクトが示されている。つまり検証系技術者が中長期的なキャリアを考えると、最初の目標がミドルレベルの検証アーキテクトであると考えられる。よってサブワーキングでは検討のベースをミドルレベルの検証アーキテクト、すなわち検証の各工程を自立的に遂行できる技術者においた。

人物像の検討においては、技術的側面、経験的側面、ビジネス的側面のそれぞれから検証アーキテクトが持つべき知識や想定される活動を考察し、求められるスキルとして整理した。検討の過程では実際の組込みシステム開発における検証の課題と期待についても意見が出され、議論に反映された。例えば第三者的視点をとるべきか、開発者視点をとるべきかについて、若干の議論が起きたが、双方に利点と欠点があり一方に偏ることはしないとの結論になった。ただし、開発の現場では意識が開発者側に寄り気味なので、第三者的視点を少し強調して扱うこととした。開発企業での検証系技術者キャリアパスや製品開発における関係者との間の役割なども実状を基に検討されている。

結果として主に以下の三点を軸として組込みシステム開発に貢献できる人材を育成するとした。

- A) ユーザの「利用シナリオ」を想像力を働かせてイメージし、テストの戦略や評価に活かせる。 実際の開発案件では設計開発側の担当者が検証を行うことも多い。開発情報から検証すべき個所を的確

#### 脚注

- <sup>\*1</sup> 「テスト設計、テスト実行等のテスト作業の実施を担当する技術者」(ETSS<sup>[6]</sup>より)  
<sup>\*2</sup> 2014年にキャリアガイドの見直しが行われ、テストアーキテクト、テストエンジニア、といった呼称は検証アーキテクト、検証エンジニアと改められた。

に指定できる利点がある一方、例外事象や想定外の操作に気づきにくいという側面がある。ユーザー視点や第三者的な視点から利用シナリオを想像することで、テスト分析の範囲を広げ、検証を設計できる能力が必要となる。

- B) 適切な手法を用いてテスト結果を分析し、製品品質に対する課題分析及び対策提案ができる。複雑化、大規模化する組込みソフトウェアの検証においても、工学的アプローチの重要性が高まっている。統計的手法やリスクベースの考え方をを用いて、効果的にテスト結果を分析し不具合原因をつきとめ対策を立てる能力が求められる。また検証結果が設計開発側やマネジメント側にフィードバックされる際、根拠や対策の有効性を工学的な視点から説明できる能力が求められる。
- C) 再発防止の観点で改善及び提案ができる。検証は個別製品の品質を確認して終わりではなく、そこで得られた知見を後続のプロジェクトに展開し組織的な品質向上に貢献することが望ましい。見落としやすい観点や技法・ツールの特徴を検証技術者間で共有したり、設計開発側にプロセス視点で改善点を提案し、同じような不具合を繰り返さない能力が求められる。

### 2.3 開発における位置付けと役割

検証アーキテクトは開発設計側や経営層へ検証視点で提案したり、各種の調整をしながら検証を進めていくことが求められる。よって開発体制においては図2のようにシステムアーキテクトと並立し、検証に関する部分を統括することになる。また、開発プロセスにおいては図3に示すように実装と単体テストを除いて開発全般に関与することになる。これらの点で、主に検証工程だけに携わる検証エンジニア(図3の点線部分)と検証アーキテクトは明確に区別される。

開発において検証アーキテクトが果たすべき役割を工程ごとに整理した(図4~図8)。図中にて検証工程に直接かかわる技術者を、実施を主に担当する検証エンジニア、プロジェクトの円滑な遂行を行う検証マネージャや検証リーダー、そして検証アーキテクトと分けた。更に設計開発側の技術者について記載し、開発全体における

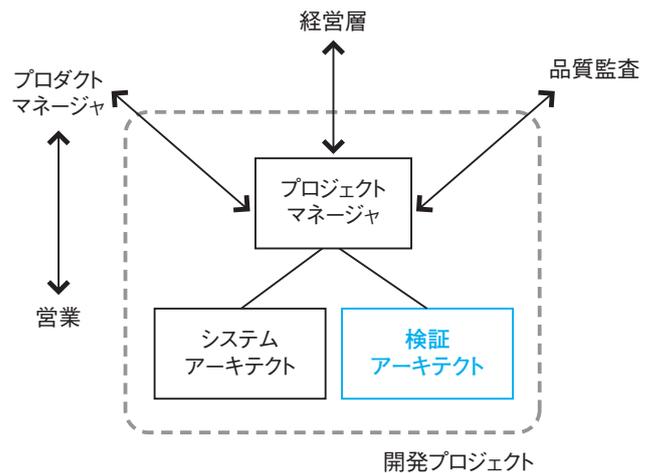


図2 検証アーキテクトの位置付け

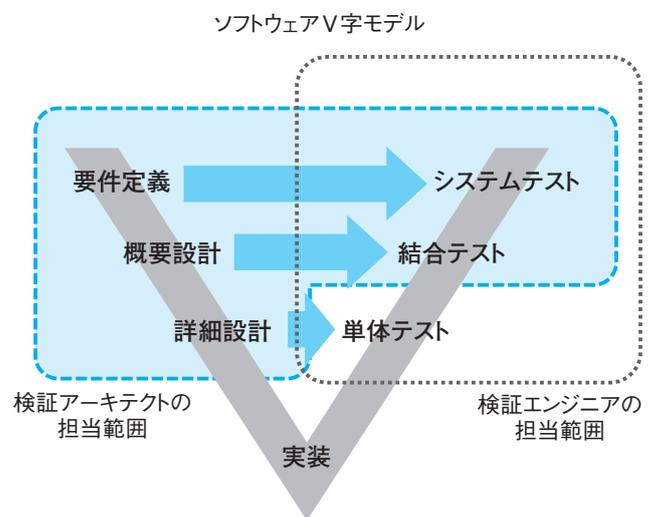


図3 検証アーキテクトの担当範囲

検証工程の各作業を位置付けた。実開発では一人の人物が検証マネージャと検証アーキテクト、またはソフトウェアエンジニアと検証エンジニア等複数の役割を果たしていることも多い。これらの区分は概念的なものであることに注意する。図中、検証アーキテクト等それぞれの技術者が行う作業を箇条書きで記載し、そのために必要とされる知識を枠を付けて表している(例:図4中の「アーキテクチャ設計技法」)。

システムアーキテクト ソフトエンジニア	検証エンジニア	検証リーダー 検証マネージャ	検証アーキテクト
<ul style="list-style-type: none"> <li>□ 企画           <ul style="list-style-type: none"> <li>・ターゲット顧客</li> <li>・製品方針</li> </ul> </li> <li>□ 要求分析・要件定義           <ul style="list-style-type: none"> <li>・要求分析・要求定義</li> <li>・実現提案</li> <li>・概算見積もり</li> </ul> </li> <li>□ システム方式設計           <ul style="list-style-type: none"> <li>・業界標準、業界規制等調査</li> <li>・知財権(特許、商標)調査</li> <li>・オープンソース調査</li> <li>・ハード、ソフト選定・調査</li> <li>・性能目標値</li> <li>・品質目標値</li> <li>・見積もり</li> </ul> </li> </ul>		<ul style="list-style-type: none"> <li>□ テスト全体計画           <ul style="list-style-type: none"> <li>・テスト方針</li> <li>・テストスケジュール</li> <li>・コスト(試算)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>□ レビュー           <ul style="list-style-type: none"> <li>・要求分析・要件定義</li> <li>・システム方式設計</li> </ul> </li> <li>□ テスト全体計画           <ul style="list-style-type: none"> <li>・テスト方針・戦略</li> <li>・テスト観点・テスト方式</li> <li>・テスト分析方針</li> <li>・品質(試算)</li> </ul> </li> </ul>
<p>フィードバック</p>		<p>フィードバック</p>	

図4 検証アーキテクトの役割(要件定義)

**要件定義:** この工程では、検証アーキテクトはシステムアーキテクトの作成した要求分析・要件定義、システム方式設定を検証観点からレビューする。また検証マネージャと連携してテスト全体計画を策定する。検証アーキテクトにはテスト観点やテスト方式を検討するスキルに加えてアーキテクチャ設計や分析に関するスキルが求められる。検証アーキテクトに期待される役割のA)「ユーザの利用シナリオをイメージする」はこの工程で果たされる。

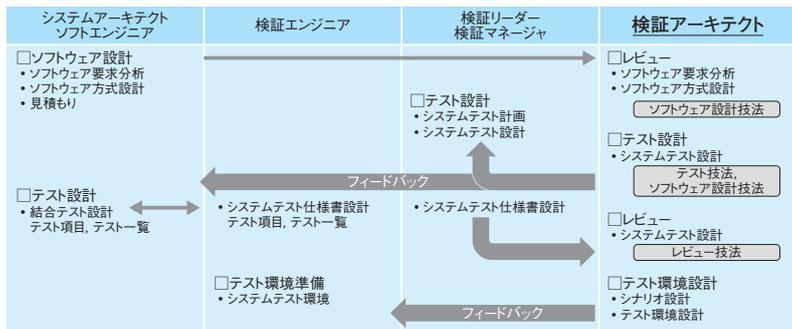


図5 検証アーキテクトの役割 (概要設計)

**概要設計：**検証アーキテクトはソフトウェア設計ドキュメントを検証観点からレビューし、システムテストを設計する。またシステムテストの環境を設計し、テスト環境や使用ツールを設計する。検証アーキテクトにはテスト技法のほかソフトウェア設計やレビューのスキルが求められる。

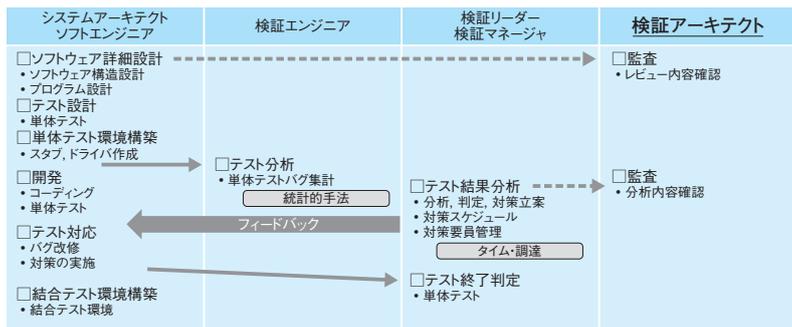


図6 検証アーキテクトの役割 (詳細設計～単体テスト)

**詳細設計～単体テスト：**この工程における作業は単体テストまで含めて多くの場合設計開発側で実施されるので、検証アーキテクトはその結果、つまり詳細設計ドキュメントや単体テストの分析結果を確認することになる。

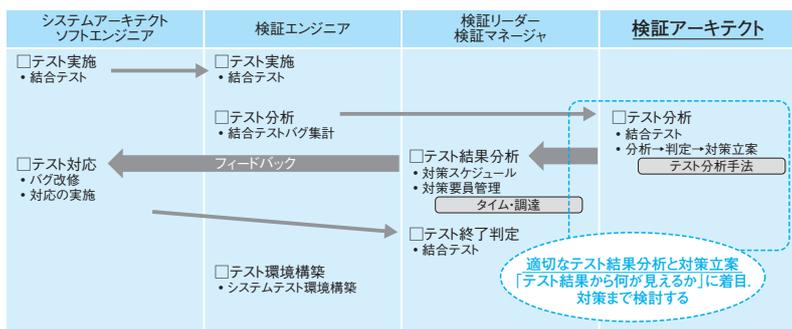


図7 検証アーキテクトの役割 (結合テスト)

**結合テスト：**検証アーキテクトはテスト結果を分析し、不具合の原因解析や対処方針の策定を行う。実施に際して各種テスト分析手法を身に付けていることが求められる。検証アーキテクトに期待される役割のB)「適切なテスト結果分析と対策提案」はこの工程と次のシステムテスト工程で果たされる。

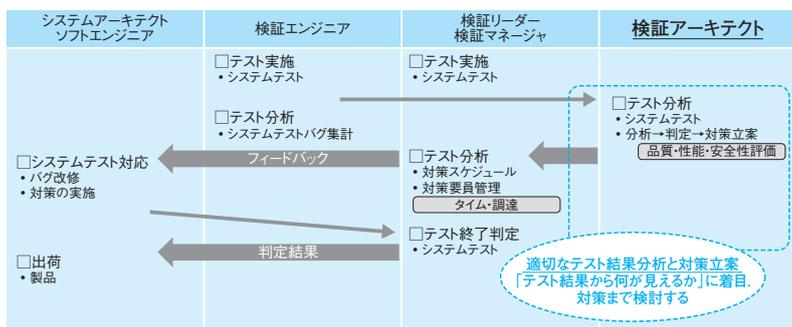


図8 検証アーキテクトの役割 (システムテスト)

**システムテスト：**検証アーキテクトは第三者的視点からシステムテストを分析する。更に不具合の原因解析や対処方針の策定を行う。実施に際して各種テスト分析手法を身に付けていることが求められるほか、品質やシステムの性能、安全性に関して判断できるスキルが求められる。

検証アーキテクトに期待される役割のC)「再発防止の観点で改善及び提案ができる」は特定の工程と関連付けることができないことに注意する。他の開発案件や開発設計者に得られた知見を展開する活動は、個々の開発案件を越えている活動であり、展開すべき注意点や改善点は開発プロセスのいずれの工程からでも得られうる。

### 3 育成カリキュラム

#### 3.1 カリキュラム策定の指針

前節で整理した人材像に照らすと、検証アーキテクトは単に知識が多ければなれるというものではなく、個々のプロジェクトに対して考察を加え、調整し、新たな発

想や視点を加えて検証を実施する能力が求められる。そこでカリキュラム策定の指針として以下の三点を設けた。

- グループ演習やワークショップを積極的に取り入れ、受講者自身が課題解決に向けて考えることを重視する。
- 事例を多く扱い、検証における課題が実開発においてどのように対処されるかを示す。
- ポイントを絞り、重要なスキルが確実に身に付くカリキュラムとする。

### 3.2 カリキュラムの講義体系

検証アーキテクトに期待される役割とスキルを核として講義で扱うべき内容を検討し、以下の4つの講義からなるカリキュラムとして整理した。組込み適塾の一部として実施することを想定し、一講義あたり0.5日或いは1日で実施する内容としている。

- [D04-01]組込み開発現場から見た検証アーキテクト(0.5日)  
カリキュラム全体の概観として、検証アーキテクトの位置付け、開発プロセスにおける役割、スキルを解説し、動機付けを行う。
- [D04-02]検証アーキテクトとしてのシステム分析・テスト設計演習(1日)  
図4に示したテスト全体計画と分析に関する講義。期待される役割A)で述べた、ユーザの利用シナリオを想像し、テストの戦略や評価に活かすためのスキルを養う。マインドマップのような発想の幅を広げる手法により、多様な視点から網羅的にテスト観点を抽出する演習を行う。また、その抽出結果を分析し質の良いテスト設計を行う演習を行う。
- [D04-03]テスト結果分析とフィードバック演習(1日)  
図7と図8に示したテストの結果分析・対策に関する講義。期待される役割B)で述べた、テスト結果を適切に分析し、対策を提案するためのスキルを養う。テスト結果のデータを分析し、事前のリスク評価を基に効果的に対処する手法を扱う。またプロジェクト中に起きたトラブルの対策をたて、関係者との調整などプロジェクトへフィードバックする演習を行う。
- [D04-04]事例から学ぶ検証アーキテクティング(0.5日)  
検証或いは開発の全体的な視点から、再発防止の観点で改善及び提案する際の知見について学ぶ(期待される役割C)。組込みシステムの開発事例において検証に関する活動から得られた知見と、それが以後のプロジェクトにどのように反映され不具合発生の予防につながったかを紹介する。

本カリキュラムの受講者は講義[D04-01]で検証アーキテクトの全体的なイメージを持ち、検証のプロセスに沿って講義[D04-02]と[D04-03]で具体的なスキルを身に付け、講義[D04-04]にて実開発での適用の際のヒントを得ることになる。

講義や演習で扱う題材、紹介する事例はハードウェア制御や組込みシステムから選び、課題や説明のポイントが組込みシステムに特徴的なものになるよう留意した。

ただし講義内容の詳細については講師の知識と経験による所も大きく、講義の趣旨やカリキュラムを講師に伝え十分に調整を行うこととした。

組込み適塾では既に講義「テスト技法」が開講されている。こちらは組込みソフト開発者が一般的に持つ知識の習得に重点をおき、本カリキュラムの前提となる講義として位置付けられる。

## 4 育成プログラムの実施と評価

### 4.1 2015年の試行カリキュラム

サブワーキングの検討を基に、2015年度の組込み適塾<sup>[8]</sup>にてアーキテクトチャ設計コース検証アーキテクティング科目として前節のカリキュラムを実施した。講義ごとのシラバスを表としてあげる。

講義名	シラバス
[D04-01]組込み開発現場から見た検証アーキテクト	1. 検証アーキテクトとは 2. 検証アーキテクトの役割 3. 開発現場の事例 4. 高信頼システム開発のための検証 5. まとめ
[D04-02]検証アーキテクトとしてのシステム分析・テスト設計演習	1. テスト計画とテストプロセス(座学) 2. テストの観点とテスト分析(座学) 3. テスト分析演習 4. テスト設計演習
[D04-03]テスト結果分析とフィードバック演習	1. 講義：テストプロジェクトの計画の立て方と管理手法 2. ワークショップ：テスト分析と対策トレーニング 3. ロールプレイ：トラブル対策会議
[D04-04]事例から学ぶ検証アーキテクティング	事例1(FPGA開発事例) 事例2(車載システム事例)

それぞれの講義につき、講義内容をよく知る有識者に講師を依頼し、扱う題材など詳細は講師とサブワーキングとの調整により確定した。例えば、[D04-02]は以下の流れで進めることとした。

- テスト設計の概要を座学で説明する。テストの戦略、設計の際の考え方(シラバス第1項「テスト計画とテストプロセス」)や、テスト設計における観点の重要性、分析手法(第2項「テストの観点とテスト分析」)を扱う。
- 分析手法としてマインドマップに注目し、発想を広げテストの観点を抽出する演習を行う。マインドマップの書き方を簡単に解説し、練習として自己紹介をテーマにマインドマップを書く(シラバス第3項「テスト分析演習」前半)。
- 現実的な題材に対してテスト分析演習を行う。受講者を数人ずつのグループに分け、自動販売機の機能仕様書を対象にマインドマップを作成する(シラバス第3項後半)。
- テストの戦略を定め、テストを設計するグループ演習を行う。前項の分析結果を題材とし、テストで確認したい品質や観点の優先度を決定する。更に、用いるテスト技法や範囲を検討しテストの計画を立てる(シラバス第4項「テスト設計演習」)。

講義 [D04-02]と [D04-03]はそれぞれ一日の講義を想定していたが、講義内容の定着と演習内容の事前把握を狙い二日にまたがって開催した。つまり講義は午後を開始し、演習課題や演習で使う技法の紹介を一日目終了までに済ませておく。実際の作業は二日目に行うので、受講者は必要ならば一日目の講義終了後に復習や準備の時間をとることができる。

## 4.2 実施の概要

検証アーキテクティング科目は、アーキテクチャ設計コースの選択科目として実施された。組込み適塾では、「コース」「科目」「講義」の三層構造でカリキュラムを整理し、受講の単位としている(下表)。検証アーキテクティング科目はアーキテクチャ設計コースの共通科目であるベース科目の履修後に、方向性を絞った内容の知識習得を目指す課程と位置付けられた。

コース	科目	講義	
アドバンストコース	システムズエンジニアリング科目	(個別講義)	
	アドバンストシステムデザイン科目	(個別講義)	
アーキテクチャ設計コース	ベース科目	(個別講義)	
	コア技術科目	(個別講義)	
	マネジメント科目	(個別講義)	
	検証アーキテクティング科目	組込み開発現場から見た検証アーキテクト	
		検証アーキテクトとしてのシステム	
		分析・テスト設計演習	
テスト結果分析とフィードバック演習			
事例から学ぶ検証アーキテクティング			
システムデザイン科目	(個別講義)		
実装エンジニアリングコース	基礎科目	(個別講義)	
	実装演習(初級)	(個別講義)	
	実装演習(実践)	(個別講義)	

受講者募集は組込み適塾全体の募集に合わせた。受講要件は講義ごとに定めたが、全体としてはテスト専門の技術者には限定しなかった。一定のソフトウェア開発経験を持ち、主にシステムアーキテクトを目指す組込みシステム分野の技術者でとくに検証に興味を持つ者一般とした。これは、設計開発と検証の両方を経験しながらキャリアを積むケースが多いことがサブワーキングの議論で指摘されたことによる。

## 4.3 実施結果と評価

受講者募集に対して6名の応募があり(内1名は特定の講義のみ受講)、カリキュラムの4つの講義すべてを2015年8月に実施した。受講者全員が組込みシステム開発企業に所属しており、主に開発を担当する技術者であった。

成績は講義ごとに評価した。課題レポートにて講義内容の理解度を評価し、また演習への参加の様子から自律的に作業や考察を進めているか、課題の達成度を評価し、

総合して100点満点で評点をつけた。結果は最低80点、最高100点、平均86.9点であり、カリキュラムの内容を伝えられたと考えられる。

カリキュラム実施結果の評価は受講者への事後アンケートとサブワーキングによるレビューにて行った。受講者アンケートについては有益性の評価、また改善のポイントについての意見集約を主眼に質問を設定した。またサブワーキングメンバーで分担して各講義を見学し、評価コメントを整理した。更に講義の担当講師を交えて改善に向けて検討した。

受講者アンケートは9つの質問を設定し、原則として各設問に対して選択式の回答と付加コメントでの回答を求めた。質問の内訳は、受講目的(1問)、講義内容の評価(5問)、その他要望・現場の課題など(3問)である。講義内容の評価では、期待への満足度、受講効果、業務への有益性と、期待通りの部分、期待外れの部分の5問を質問した。図9～図11に示すように、科目全体としては受講者全員から肯定的な評価が得られた。更に「期待通りの部分」については、「テスト計画の概念をきちんと知ることができた」「すぐに活用できる情報が入手できた」として[D04-02]、[D04-03]の講義内容が挙げられた。一方「期待外れの部分」については「業務に反映させるところが見えない」「内容が多岐にわたり消化しきれない」との意見が出たものの、特定の講義や内容に回答が集中することはなかった。現場の課題へのコメントで「テストの方針決めて困っている」「計画が不十分」といったものが得られている。総じて検証アーキテクトのスキルが開発現場の要望から外れておらず、カリキュラムの方針は適切であったと考えられる。

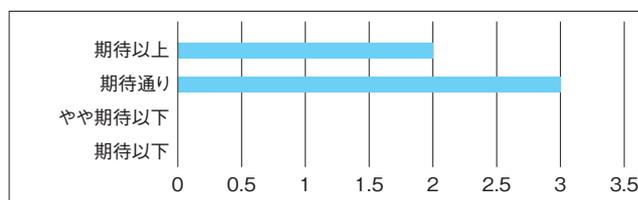


図9 検証アーキテクティング科目は全体として「期待通り」であったか

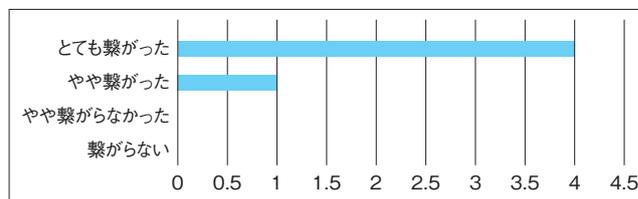


図10 検証アーキテクティング科目はスキルアップにつながったか

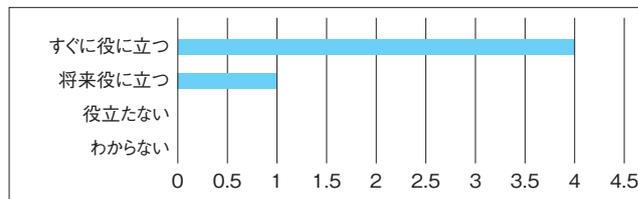


図11 検証アーキテクティング科目は自身の業務に役立つか

サブワーキングによるレビューでは、一つの講義あたり2～3人のメンバーが講義を見学してレポートを作成した。観点を以下のように設定し、各講義の章ごとに評価コメントと10段階の評点をつけて評価を行った。

- 講義の題材・講義内容が人材像に合致しているか
- 講義・実習方法が効果的か
- 受講生の興味・活性度
- その他見学者からのコメント

すべての講義が終了した後、レポートを基にサブワーキングで総合的に評価を行った。例えば講義 [D04-02] に対する議論では以下の点が指摘された。

- 内容はサブワーキングの議論に沿っており問題はない。
- 当初、受講者の反応が堅く議論への積極性が足りない様子。座学が続き受講者間のコミュニケーションが不十分なことが原因と思われる。
- マインドマップの作成で受講者が迷っていた。技術の習得には試行錯誤が重要であるが、適当なタイミングで助言してはどうか。
- 演習題材(仕様書)の分量が多く、その理解に時間がとられているように見えた。
- 演習を進めるグループが一つだけだったので、アプローチの広がりや比較が不十分であった。

サブワーキングでの評価においては、資料の多さや講義間で説明の繰り返しが生じたことにより、内容と時間のバランスが適切でない講義が幾つかあったことが指摘された。講義で扱われる技法や事例は3.2節で述べたカリキュラムに沿ったものであるが、それらの意義や検証アーキテクトの視点、考え方にまで考察を深める余裕を受講者が持てなかったことが懸念される。講師担当者と十分に調整し、カリキュラムの特徴を伝えられる題材の選択、議論や総括のための時間確保といった対策により、育成プログラムの効果が高まると考えられる。

講義実施に対し、ほかにも以下の表に挙げるような改善事項が指摘された。それらへの対策も併せて検討されており、関連する資料の作成や講師へのフィードバックとして対応がとられている。

改善すべき点	対策
講義間の内容に重複がある、用語が統一されていない。	事前に講義内容を確認し調整する。
講義間の関連や講義内容の位置付けが受講者に伝わっていない。	目指す人材像や役割に関する資料を作成し、関係者(講師、受講者など)に早い段階で説明する。
グループワーク、議論の活発性が不十分。	講義冒頭でアイスブレイクを行う。多くの受講者を確保する。

## 5 総括

組込みシステム開発の検証系人材育成カリキュラムを開発し、組込み適塾にて実施した。カリキュラム策定に当たっては開発企業の要望・意見を取り入れ、開発全体の視点で検証を統括する検証アーキテクトの育成を検討のベースにおいた。実践的なスキル習得を目指して演習や事例を重視し、4つの講義からなるカリキュラムを定めた。

検証アーキテクトとして実開発に貢献していくためには、多くの事例や経験を基にして自身の検証に関する知識を見直し、実開発に適用できるものとして定着させる必要がある。本論文で紹介したカリキュラムは検証アーキテクトに求められる技術や視点を扱っており、検証技術の向上に有益だと考えられる。しかし知識の見直しと定着はカバーできていない。講義期間終了後に質疑・相談の機会を設ける、事例を紹介するなどのフォローアップにより育成プログラムとしての質の向上が期待できる。

試行にて一定の有効性が確かめられたため、次回の組込み適塾から本格的にカリキュラムを実施する予定である。カリキュラムの内容、実施体制などについて挙げた改善点については既に検討が進んでおり、今回は更に効果の高い講義を実施する。本カリキュラムにより高度な検証技術者が育成され、組込みシステムの品質や信頼性が向上することが期待される。

## 6 謝辞

本育成プログラムを組込み適塾の中で企画実施する際にご協力いただいた、検証系人材育成サブワーキングの皆様、教育事業部会長前川隆昭様並びに部会の皆様、組込み適塾塾長井上克郎先生に感謝いたします。またカリキュラム内容の検討に当たり宮崎大学工学教育研究部片山徹郎先生に多くの助言をいただきました。ここに記して感謝いたします。最後に本原稿に関し有益で的確なコメントをいただきました査読者の方々に感謝いたします。

### 参考文献

- [1] 東田光裕, 岩井匡代, 八木浩, 奈良木英人, “組込みシステム産業振興機構の紹介,” SEC journal, Vol.9, No.3, pp.150-151, 2013.
- [2] 井上克郎, “組込み技術者向けキャリアガイドの開発,” SEC Journal, Vol.8, No.2, pp.85-88, 2012.
- [3] ISO/IEC/IEEE, 29119-2 “Software and systems engineering - Software Testing Standard -Part2:Test processes”, 2013.
- [4] B.Rollison, “How We Test At Microsoft マイクロソフトでどのようにテストをしているのか?,” JaSST '12 Tokyo, 2012.
- [5] A. Page, K. Johnston, B. Rollison, “How we test software at Microsoft (R)”, Microsoft Press, 2008.
- [6] 組込みソフトウェア管理者技術者育成研究会, 情報処理推進機構ソフトウェアエンジニアリングセンター, “ETSS標準ガイドブック”, 日経BP, 2006.
- [7] JSTQB, <http://jstqb.jp/index.html>.
- [8] 第八回組込み適塾講座一覧, <http://www.kansai-kumikomi.net/ptraining/8th/index.html>.