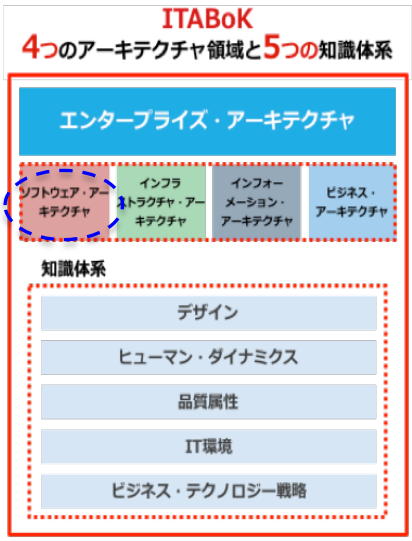
第９回　コラム　「ソフトウェア・アーキテクチャの専門領域」

Iasa 日本支部 梶川　哲生

このシリーズの第９回目は、ITABoK(Version2)の第２章＜ケイパビリティ・ガイドブック＞の後半で取り上げている４つの専門領域の中の**「ソフトウェア・アーキテクチャの専門領域」**を取り上げます。

最初に、ITABoK Version２に於ける「ソフトウェア・アーキテクチャの専門領域」の全体の中での位置付けを確認しておきたいと思います。

右図は ITABoK 全体の構成を鳥瞰図に表した図です。



上部の４つの専門領域の一つとして「ソフトウェア・アーキテクチャ領域」が位置付けられています。（右の図の青い点線部分）

既に他の２つの「専門領域」と図の下部の「５つの知識体系」は既にこのシリーズでご紹介した領域になります。今回の「ソフトウェア・アーキテクチャの専門領域」では、その知識/専門領域の主なカテゴリーの理解と、専門領域間の共通性を比較対照する能力を以下の７項目について説明していますが、今回は特に**太字**の３つについて取り上げます。

ソフトウェア・アーキテクチャケイパビリティ

- **開発、方法論、プロセス**
- ソフトウェア・アーキテクチャのツール
- **アーキテクトのためのソフトウェア・エンジニアリング**
- サービス、ワークフロー、メッセージング
- **高水準の品質属性**
- テクノロジー、プラットフォーム、フレームワーク
- データ/情報/知識の管理

（ITABoK V2 日本語訳 印刷版 P351, デジタル版 P412 より）

ITABoK がソフトウェア・アーキテクチャの専門領域の中で注目すべき箇所を取り上げる前に、先ずソフトウェア・アーキテクチャの検討段階では**そもそも何の目的で行うのか**を確認しておきたいと思います。

私達はこのソフトウェア・アーキテクチャを検討する過程に於いて、特に特定のプロジェクト案件を視野においている場合は、非公式な形で特定のソフトウェア・ソリューションやパッケージ・ソリューション、または、その組み合わせをイメージする方も多いのではないでしょうか？
しかしながら、アーキテクチャの「プロセスと成果物」を定義する事を目的とする代表的なアーキテクチャ・フレームワークの一つである「TOGAF Verion9（日本語訳)」では、「第11章　情報システム・アーキテクチャ　- アプリケーション・アーキテクチャ」でアーキテクチャの設計の目的について以下の様に述べています。

> ここでの目的は、データ（インフォメーション）・プロセスとビジネス・サポートに必要な主要なアプリケーション・システムを定義する事である。
>
> ここでは、アプリケーション設計の作業に触れるのが目的ではないことに注意しておくことである。ゴールは、どんな種類のアプリケーション・システムがエンタープライズに関係するのか、それらのアプリケーションがデータを管理するために、そしてエンタープライズ内の人やコンピュータ・アクターに対して情報を提供するために何をする必要があるかを定義することである。
>
> アプリケーションは、コンピューター・システムとして記述されるのではなく、データ（インフォメーション）・アーキテクチャのなかのデータ・オブジェクトを管理し、ビジネス・アーキテクチャの中のビジネス機能をサポートするケイパビリティの論理的なグループとして記述される。
> 「TOGAF Version9 日本語訳版 P127 より。括弧内（ ）は ITABoK 内の用語」」

上記の下線部分の記述にある様に、この段階ではアプリケーションの設計作業に触れるのが目的ではない事、またインプットとしてデータ（インフォフォメーション）・アーキテクチャとビジネス・アーキテクチャの情報を前提としている事が示唆されています。

従って、ソフトウェア・アーキテクチャの検討では、既に紹介した他のアーキテクチャ専門領域の確実な理解も必要になります。通常のプロジェクト案件ライフサイクルの視点で見ると、特定の実装やソリューションに依存しない論理構造や論理コンポーネントを分析・定義した上で、物理構造や物理コンポーネントに落とし込むプロセスは時間の制約などで冗長に見える場合が多々ある事でしょう。ただ同時にこの様な、特定のソリューション在りきの導入プロセスの積み重ねが招く結果は容易に想像がつくところです。目指すべアーキテクチャが維持されないのです。

さて、ソフトウェア・アーキテクチャ　ケイパビリティで着目する１つ目は、**「ソフトウェア・アーキテクチャ開発の方法論とプロセス」**についてです。

概要

手法の主な焦点（アーティファクト、ユースケース、属性、ドメイン）に関係なく、アーキテクチャのデザイン・プロセスには以下の観点が含まれます。

１．アーキテクチャ要件の分析、関心事の特定、及び、プロセスを通した意思決定への背景の提供
２．システムとその進化の望ましい特性と補助的原則の文書化、及びシステムのステークホルダーへの、それらの伝達の促進
３．一貫した方法での、アーキテクチャの代替案の評価と比較
４．実際のシステムの導入がアーキテクチャ記述に準じているかどうかの確認

（ITABoK V2 日本語訳 印刷版 P352, デジタル版 P413 より）

ここで、アーキテクチャ要件の分析については以下の様に説明しています。

アーキテクチャ要件の分析とは、システムのゴールと期待される品質属性を特定するプロセスです。その要件はビジネスやミッションのゴールと直結し、システムのステークホルダーと明確に関与していなければなりません。分析で最も重要なステップの１つは、デザインへの関連（最終的に実現されるシステムが現在の運用環境に適合する場所）を明確にする事です。

（ITABoK V2 日本語訳 印刷版P353, デジタル版P414より）

アーキテクチャ要件の分析には、期待される品質属性の特定や現在の運用環境への適合評価を明確にする活動が含まれている事に注目します。

更に、アーキテクチャの代替案の分析については、いくつかの評価手法によって潜在的なリスクを特定する事だと説明しており、評価の手法としてソフトウェア・メトリクス、シナリオ、属性モデルベースの３種類を簡単に紹介しています。

アーキテクチャのコンプライアンスの評価の説明には以下の記述があり、読者の中には過去に思い当たる点がある方もいらっしゃるのではないでしょうか。　これらの逸脱は特に何らかのガイドラインが存在しない場合には、放置され蓄積され続けるでしょう。

システムの導入と進化の期間中、たとえ最初のリリースの場合でも、定義されたアーキテクチャからの逸脱を観察するのは一般的な事であり、それらの逸脱は開

発者の認識の欠如、矛盾する要件、遅い段階での要件の変更、技術的困難や締め切りや予算上のプレッシャーに起因しています。このような逸脱は大抵の場合時間と共に蓄積され、アーキテクチャの浸食として知られる現象を招きます。
（ITABoK V2 日本語訳 印刷版P354, デジタル版P416より）

更にアーキテクチャのコンプライアンスについては、「ソースコードで実装されたアーキテクチャが定義されたアーキテクチャに準じる程度を測るための評価基準です。」と説明しています。実践に於いては、アーキテクチャのコンプライアンス準拠のレベルを個別のプロジェクト案件で実装された各ソリューションに対して評価を行い、特定されたギャップは記録した上で必要な対処を検討する事になります。

２番目に着目した点は、**「アーキテクトのためのソフトウェア・エンジニアリング」**についてです。

概要

《あらゆるエンジニアリングの活動と同様に、ソフトウェア・エンジニアリングは「問題の定義付け」で開始され、「問題へのソリューション」を獲得するためにツールを適用します。しかし他のエンジニアリングとは異なり、ソフトウェア・エンジニアリング（SWE）はソリューションを獲得するための方法論や手法に重点を置いているように思われます
（ITABoK V2 日本語訳 印刷版P357, デジタル版P418より）

ここでは「ソフトウェア・アーキテクチャ」と「ソフトウェア・エンジニアリング」の主たる違いについては以下の様に説明しています。

Iasa の定義では、アーキテクチャとはテクノロジー戦略の応用を通じた、ビジネス、組織、クライアントの利益の獲得の実践であり、目指すのはビジネスに対する価値の実現です。他方、ソフトウェア・エンジニアリング（SWE）は特定の目的のソリューションを獲得するための方法論や手法に重点を置いている、と説明しています。実際の検討段階では双方の異なる視点から同じ対象や問題点についての分析や評価もある事でしょう。しなしながら、その本来の目的と重点が異なると解説しています。

3番目に着目した点は、**「高水準の品質属性」**についてです。

説明

品質属性（QA）は非機能的な要件ですが、機能的要件なしにはそれを説明する事は出来ません。要件は同一であるにもかかわらず、品質属性に対する異なるフォーカスによってソフトウェア・アーキテクチャ内の差異が生じるため、これらの「非機能的」要件を「特別な機能」要件と呼ぶ事には大きな意味があります。

概要

要件は、アーキテクチャ的に優れたソフトウェアを作成するのに十分なものではありません。異なるアーキテクト達は、同一の要件に対して異なるアーキテクチャを作成するでしょう。

（中略）

アーキテクトにはステークホルダーからの情報を獲得する必要があります。ステークホルダーは、大抵要件の観点から物事を考慮しますが、アーキテクトは現在の実装と将来の拡張の観点から思考しなければなりません。ステークホルダーの主たる関心事はコストですが、アーキテクトの主な関心事は不可欠なQA（注：品質属性）を特定して、それらをステークホルダーが満足するように導入する事です。

（ITABoK V2 日本語訳 印刷版P379, デジタル版P443より）

これらを実践するにあたっては、ITABoKではステークホルダーをまじえた品質属性ワークショップ（QAW）、品質属性シナリオ、品質属性のテスト、パフォーマンス、セキュリティ、品質属性の評価、品質属性の経済分析などを通じ、高水準の品質属性を確保する事が可能となると説明しています。高水準の非機能要件を長期に亘りアーキテクチャで担保するには、それなりの工夫、努力と活動の継続が必要であり、その上に実装されるソリューション群の安定性と必要となる多様なパフォーマンスを発揮できるアーキテクチャ基盤の維持に貢献する事に繋がります。

（ITABoK V2 日本語訳 印刷版P379～382, デジタル版P444～447より）

■　終わりに

今回は、「ソフトウェア・アーキテクチャの専門領域」の中の３つのケイパビリティについて取り上げました。ソフトウェア・エンジニアリングと被るトピックも多い領域ですが、アーキテクトはエンジニアリングの知識や知見も活用しながら最適なソリューションを導き出し、該当システムの長いライフサイクルに亘って目指すべきアーキテクチャ（ケイパビリティの実現）を維持する責務を負っている点が他の各専門的エンジニアと異なる役割と言えます。

このコラムに関して皆様からのご意見や感想、参考になる事例等がありましたら是非お聞かせ下さい。本コラムが ITABoK の理解とこれからの IT アーキテクトのあるべき姿の一助になる事を願っています。

＜終わり＞

参考：

アーキテクチャの策定プロセスや成果物（アーティファクト）の業界標準の一つとして、Iasa では TOGAF を引用・参照している経緯があります。

TOGAF (OpenGroup)について

http://www.opengroup.or.jp/togaf.html

ArchiMate (OpenGroup)について

https://en.wikipedia.org/wiki/ArchiMate

http://www.opengroup.or.jp/archimate.html