論文執筆要領A

－論文・技術報告テンプレート　－

チーム№：チーム名：リーダー名

How to write Paper

－　Format Template　－

Team’s name：Leader’s name

この資料は，「JGS研究プロジェクト論文」の形式・構成・内容に関するガイドラインである．入力用のテンプレートを兼ねており，当資料の文章等を置き換えていけば提出論文の形式が作成できるようになっている．書き出しの日本語Abstract（要旨）部分はこのフォーマットのように書き，英語のAbstractおよびキーワードを次のように記述する．

This manual guides you in writing an “JGS Paper” in the proper format and structure. The guideline is prepared as an input format template to help you write a properly formatted paper by following the instructions explained in the body. The abstract of your paper should be written in this style, with five key words and phrases as shown below.

Key Words & Phrases ：執筆要領，論文テンプレート，発明提出，フォーマット，参考文献

　　　　　　　　　　 paper guideline, paper template, invention submission, format, references

１．はじめに

日々新たなデジタル技術が開発されている昨今，IT業界のみならず様々な業種でデジタル技術を駆使したビジネスが創出され，企業価値を高めている．このような状況の中，我が国では更なる競争力強化のため，各企業にデジタルトランスフォーメーション(DX)を推進していくことを強く求めているが，現状では一部の先進的な企業を除く多くの企業がDXを推進できていない。その主な原因として，既存の基幹システムの老朽化が挙げられている．年を重ねることでシステムが複雑化することで，維持管理費が高騰，また技術継承もままならなくなり，新たなデジタル技術をビジネスに組み込めない状態となっている． 経済産業省のレポートによれば，レガシーシステムが継続して使用された場合，2025年～2030年の間に最大12兆円の経済損失が生じると推定されている[1]．

そのため，DXを推進していくためには，老朽化し複雑化したレガシーシステムの改修が必要となる．しかし既存システムの改修は大規模かつ長期のプロジェクトとなるためリスクも大きく，スピーディに効果を得ることができない．そこで，頻繁に改修が求められるビジネスロジック又は新規に作成するビジネスロジックについて，新たなデジタル技術を使用して構築することで，ビジネスモデルの変化に迅速に追従できるようにする必要がある．

近年，維持管理費を削減し，ビジネスアジリティを向上させるアーキテクチャとしてサーバレスアーキテクチャーが注目されている。サーバレスアーキテクチャーは，各クラウド事業者の責任範囲が広く，ユーザは業務ロジック構築とアプリケーション開発に注力できる点が大きな特徴である．以下，2章でサーバレスアーキテクチャーの特徴を述べ，3章，4章にてサーバレスアーキテクチャーを適用することで，ビジネスアジリティが向上するかを作成する成果物の量及び開発作業量の2点から検証を行う．

２．サーバレスアーキテクチャーとは

　サーバレスアーキテクチャーとは，一般的に「サーバーを必要としないアプリケーション実行環境を組み合わせたシステム構成」を示すが、 本論文では「サーバレス」，「サーバレスアーキテクチャー」の２つの用語は、2019年度JGS論文「サーバレスアーキテクチャーの適用検討プロセスの提案」[2]の定義と同意とする．

2.1 サーバレスの特徴

　サーバレスには, 各クラウド事業者ごとに様々な特徴が存在するが，ここでは代表的な特徴について述べる．

1. サーバーの構築・運用が不要

サーバーやランタイム等のアプリケーションの実行に必要な基盤環境については，図１[xx]の通り、クラウド事業者の責任の下，運用・管理されている．そのため、環境構築を行う必要がなく、運用・管理に掛かる作業負荷も大幅に削減される．



図１．サーバレスの責任モデル

1. 柔軟なスケーラビリティ

処理量に応じて，自動的なスケーリングが実施可能．

３．サーバレスアーキテクチャー適用によるビジネスアジリティ向上の検証

本章では，1章で述べたサーバレスアーキテクチャーの適用効果測定する．サーバレスアーキテクチャーを使用してグッズ購買サイトを構築し，オンプレミス環境における作業と比較することで，ビジネスアジリティが向上することを検証する．

なお，本論文ではビジネスアジリティの向上とは，サービスリリースまでの期間が短縮されることを指す．

検証観点は以下のとおり．

・ドキュメント作成量

・構築作業時間

3.1仮説と前提

3.1.1.仮設

サーバレスアーキテクチャーを適用することで，各開発工程のドキュメント作成量削減，構築作業時間が削減され，開発期間が短縮することを仮設とする．

3.1.2.前提

(1)利用するクラウド事業者

本検証では，代表的なクラウド事業者[4-1]であるAmazonWebServiceの各サービスを用いる．

1. モデルシステム構築

検証用のモデルシステムとして，東京オリンピック2020オリジナルグッズ購買システムを構築する．同システムに必要な要求事項を要求仕様書（別紙）として纏め，要求を充足するシステムを構築する．適用するサーバレスアーキテクチャーは，Serverless Airline Booking[4-2]を参考に検討を行った．システム構成は図xに示す．

3.2検証方法

3.2.1構築作業時間

要求仕様書を基に，採用するAWS各サービスで必要となる設計および設定の検討を行い、パラメータシート（別冊x）に纏める．同シートに沿って，AWSコンソール，AWS CLIを利用して構築作業を実施する．

3.2.2ドキュメント作成量

オンプレミス環境でウォーターフォールモデルによる開発を行う場合に必要となる成果物一覧を作成する．3.2.1の構築作業結果より，サーバレスアーキテクチャー適用による開発で必要となる成果物を記載し，比較・評価の結果を評価シート（別冊x）として纏める。

3.3検証結果

3.3.1 構築作業時間

AWSサービスの構築時間：10時間

（うち，Lambda関数コーディング：7時間）

3.3.2ドキュメント作成量

オンプレミス開発68本，サーバレス開発56本．サーバレス開発ではオンプレミス開発との比較でドキュメント作成量が18%減少した．サーバレス開発時のドキュメント1本あたりの作業ボリュームについて，以下に記載する。

・オンプレミス同等のボリューム38本

・オンプレミスよりボリューム減少18本（サーバレス開発全体の32%）

ボリューム減少理由は，サーバレス開発では，利用者は要求事項に合わせて，提供される機能を利用することを選択し，設定値を決めれば良い．設定値に沿った構築作業はクラウド事業者が担うためである．

４．考察

3.3検証結果より，サーバレスアーキテクチャーを適用することで開発期間の短縮が可能であり，ビジネスアジリティの向上が確認できた．開発期間が短縮された主たる要因は， [参照]評価シートより，非機能関連のドキュメント作成量および構築作業時間が大きく削減されたことである．一方で，本検証では，業務ロジックおよびアプリケーション開発に対しては，開発期間の短縮効果は確認できなかった．

５．サーバレスアーキテクチャー適用おける課題

今回の検証結果より、次のような課題が認められた。

**最適なサーバレスアーキテクチャーの選択**

　本検証では、モデルケースのアーキテクチャーを採用したが、他のAWSサービスの組み合わせでも実現可能であった。

クラウド事業者より、多くのサービスが提供されているかつ、各サービスを組み合わせたユースケースも公開されており、システムの特性に応じたサービスおよびユースケースの選択が必要である。

**開発言語の選択**

　Lambda関数構築時に、開発者の得意な言語をそれぞれの関数で選定したため、複数言語が混同したシステムとなった。

　AWSのLambdaでは、複数の言語をサポートしているため、開発工程前にコーディング規約を定める必要がある。

**クラウド人材の育成**

　AWS未経験者にて本検証を行ったため、構築作業前の事前学習に延べ１ヶ月程度費やした。（AWSの基本概念、各サービス仕様等）

　AWSサービスは日進月歩で進化しており、エンタープライズレベルの開発を行うためには、クラウド人材の育成が必須となる。

**運用・監視の複雑化**

　検証作業にて、エラー原因を特定するため、ログの解析を行ったが、ログ。しかし、上記にも記載の通りAWS未経験者で作業を行ったため、

各サービスの稼動状態の監視とログ確認は、個別サービスごとに必要になるため、システム全体での

運用における各々のサービスの連携に関してエラーが追いづらくなる為従来のシステムのような統一的な運用が難しい

６．今後の展望

サーバレスアーキテクチャーを用いることで、開発時に作成するドキュメント量と開発作業量が削減され、その分アプリケーション開発へ注力することが可能であることがわかった。 また、今回検証できていない以下の２つの観点を追求することで運用コストの削減と更なる高速開発を実現できる可能性を秘めている.

１つ目は、監視運用である.クラウド事業者からは、様々な監視サービスが提供されており、これらサービスを組み合わせて使うことで、障害時の自動復旧や日々のモニタリングよる障害の未然防止などに役立つ。

２つ目は、各サービスの疎結合化である.各アプリケーションをマイクロサービスとして独立させることで、仕様変更時にも対応範囲を局所化し、 低コスト・短納期で開発が可能となる.さらにドメイン駆動設計やアジャイル開発と組み合わせることで、更なる効果を期待できる.

ただし、これらを実現させるためには、高度な技術力とクラウドに対する豊富な知識が求められるため、企業としてはクラウド人材の育成が今後の課題である.

参考文献　　　　(記述例，[3]は架空のもの)

1. 大野侚郎，「論文作法」基礎の基礎，2002年度IBMプロフェッショナル論文発表会講演社内限定資料，2002年11月
2. 木下是雄，理科系の作文技術，中公新書，ISBN4-12-100624-0，1981
3. 技術太郎，ペリルの分類を利用したリスク事象特定法，XX学会誌，Vol.5,No.3，pp.25-30，2003
4. PMI Standards Committee, *Project Management Body of Knowledge-2000 Edition*, Project Management Institute, Newtown Square, Pennsylvania, ISBN1-880410-23-0, 2000
5. IBM Japan, http://www.ibm.com/jp/, 2003.2.28

添付１：メンバーリスト ( 会社名アイウエオ順 )

添付２：Member list

(alphabetical order of company name )添付１：メンバーリスト ( 会社名アイウエオ順 )

添付２：Member list

(alphabetical order of company name )

