形式：起承転結

タイトルスライド

1. 紹介
   1. 研究内容・結論 or 疑問を提示するまで

サーバーレスについては、良いことばかり言われているが、

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のは本当か？

1. 問題意識の提起
   1. DX事例紹介

DXに取り組むことでビジネスの競争力維持・強化につなげる。

事例を取り上げる。

先行する企業に対抗するために、何に取り組んでいけばよいかの

* 1. 現状紹介

DXに取り組む前の現在（レガシー）について説明

* 1. 対策の紹介

レガシーシステムを切り崩す新たなデジタル技術としてサーバレスが注目されている。

1. サーバレス、サーバレスアーキテクチャー定義
   1. 用語解説

サーバーレス　の定義

サーバレスアーキテクチャー　の定義

* 1. ユーザ／クラウド事業者の責任モデル

サーバーレスの責任モデル

　オンプレ、IaaS、PaaS、サーバレス

1. 検証
   1. 仮説

開発期間が短縮、観点は以下の通り

　 ・構築作業時間

　 ・ドキュメント量

* 1. 検証方法

ビジネスアジリティの向上をどのように検証するのか？

サーバー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ました。

オンプレ開発と比較して開発期間が短縮されるか検証した。

* 1. 検証作業

・事前学習の苦労話も伝えたい

* 1. 検証結果

1. 考察
   1. 非機能要求部分の実装自動化

サーバレスには拡張性及び可用性を向上させる機能が組み込まれている．ユーザーは要求事項に合わせて，サーバレスの各サービスが提供する機能（バックエンド接続，拡張性，バックアップ等）を選択し，設定値を決めれば良い．加えて，クラウド事業者で実行が担保されており，ユーザーの設計範囲が限定されるため，オンプレミス環境で開発時より非機能部分の構築作業時間とドキュメントの作成量が削減される．

* 1. 関数が実現する機能の簡素化

各Lambda関数が「目的が1つもしくは2つで，ステートレス」な設計となっている

* 1. 結果

サーバレスアーキテクチャーを適用することで，サービスリリースまでの開発期間が短縮され，迅速にサービスの提供が可能になる．結果，ビジネスアジリティが向上すると言える．

* 1. サーバレスアーキテクチャーの適用により作業量が減少しない作業について

業務要件定義や，アプリケーションの構造・設計，利用するデータの設計はユーザーの責任範囲である．

1. 課題
   1. 開発言語の選択

適切な言語を選択する自由度が増す反面，保守性が低くなるため，開発工程前に言語を選定するのが望ましい

* 1. クラウド人材の育成

1. 今後の展望

ビジネスアジリティを更に向上させる可能性がある観点は以下です。

* 1. 開発業務への注力
  2. アプリケーションのマイクロサービス化