FUJITSU

生成AI×富士通実践知で柔軟・高品質なモダナイズ

Fujitsu PROGRESSION

2026年7月

富士通株式会社

Fujitsu PROGRESSION

お客様課題	● 現行アプリケーションのビジネスロジックを維持しつつ、 稼働レガシー言語を、短期に・容易にオープン化/最新化したい
提供機能	● リライト機能：COBOL→Java自動変換 ● リホスト機能：COBOL→NetCOBOL自動変換 ● Fujitsu PROGRESSION x 生成AI
特長/強み	● 富士通グループによる自社開発 ● 移行先フリー/ベンダーロックなし ● リライト/リホスト比較による最適選択 ● 国内外で多数の実績

レガシーシステムのモダナイゼーション

FUJITSU

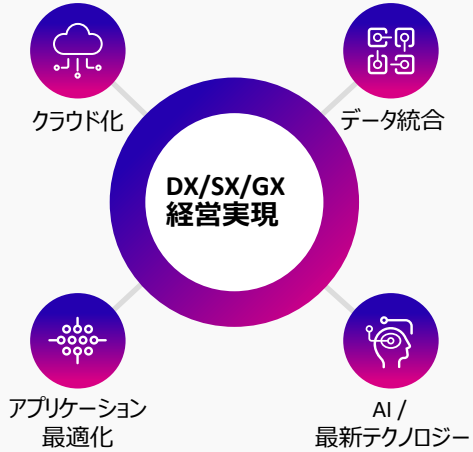
From
既存資産

MF/UNIX
レガシー資産

モダナイゼーション

Fujitsu PROGRESSION&生成AI

To
DXシステム/新基盤



- データ利活用・連携限定
- 先端IT活用立ち遅れ
- 業務改革/経営刷新の限界
- サステナビリティ経営への不安
- レガシー維持コストUp

- レガシースキル人材不足
- 仕様書・設計書不在
- 最適な移行選択肢不明
- 適正コスト・期間不透明
- 肥大化・複雑化・ブラックボックス化

- 先端ITの利活用
- データドリブン化
- 業務改革/経営刷新の推進
- サステナビリティ経営強化
- レガシーコスト脱却/競争投資へ

リホスト・リライトサービス：Fujitsu PROGRESSION

最適解を選択可能

- **JavaへのリライトとCOBOL環境へのリホスト、**２つのモダナイズ方式に対応
- 資産の状態やお客様の方針に応じて、**最適なモダナイズ方式**をご提案

富士通グループソリューション

- 様々な業種でのシステム構築で培った**SI技術を組み合わせてサービス提供**
- 富士通グループの開発製品のため、**迅速かつ細やかな技術サポートが提供可能**

国内外での多数の実績

- 富士通メインフレーム**「GS21シリーズ」**に対応
- 国内外メインフレームを中心に、**20年以上**にわたり、多くのお客様の移行に成功
  リライト：海外中心に**50社 /** リホスト：国内中心に**90社**

Fujitsu PROGRESSION
リライトサービス概要

Fujitsu PROGRESSION リライトサービス

業務ロジックを維持した止まらないモダナイゼーションを実現
富士通独自開発のリライトサービス

① 現行資産の構造を維持して Java言語へ変換

② AI活用の道を拓き モダナイ期間短縮・保守性向上

③ DXを意識した拡張性を 持つ富士通のリライトサービス

※ PoCで適用可否を見極めながら段階的にモダナイゼーションを進めることも可能です

①現行資産構造を維持した高品質Java変換

リライト

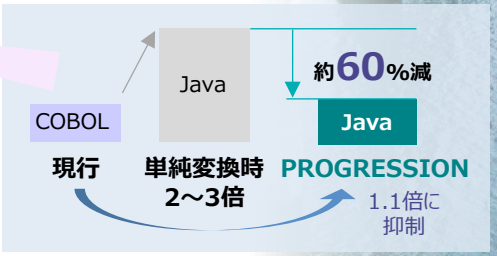
FUJITSU

特 長　COBOL特有のロジックを共通化した変換方式

- **お客様の作業負荷を最大限に軽減**
  COBOLからJavaへの1対1変換を実現。業務仕様・処理ロジックを維持して作業負荷を軽減
- **安全なモダナイゼーションを実施可能**
  全面刷新ではなく、まずは“止めない・壊さない”モダナイゼーションが可能

優位性　独自フレームワークで肥大化しないJava変換を実現

- **過去20年間のCOBOL変換実績を基にした富士通独自フレームワーク**
  変換後のソース規模を最大約60%削減（当社実績比）
- **ソースコード変換時、高いテスト効率で工期圧縮**
  高い機能踏襲性により、問題発生時の原因箇所特定が容易



PROGRESSIONリライトサービスの特長と優位性

②DXへの拡張性を持つリライトサービス

リライト

FUJITSU

特 長　将来の拡張を見据えたリライトサービス

- **Java化により、リソース配置の柔軟性が向上**
  コメントを含んだJava変換で可読性を高め、COBOLとJavaの経験者間での技術継承を容易に
- **ベンダーロックインの排除と柔軟性のあるシステム拡張が可能に**
  フレームワークをオープン化することにより、移行後も拡張が容易

優位性　モダナイゼーション市場トップシェアの実績と実践知を反映

- **Fujitsu PROGRESSIONで国内外50社以上の対応実績**
  COBOL特有の構造やお客様運用を理解した変換設計により、実行可能性を重視
- **富士通による実践知と実行力**
  多数のミッションクリティカル領域の移行経験と実績により、“納得感ある進め方”で実現

Fujitsu PROGRESSION リライトサービス 【MF】

- アプリケーションのビジネスロジックには手を付けずに、メインフレーム資産をオープン環境に適したJava等の言語へと変換いたします。
- JCLやAIM/NDB、VSAMなどの移行（定義体変換）にも対応しております。

※：JakartaEE対応のFujitsu Software Enterprise Application Platform製品

リライトサービスを活用したモダナイゼーションの流れ 【MF】

- 対象資産特定・PoCにて、移行対象資産を確定し、本サービスの適用可否を判断します。
- 後続の設計工程で技術要素を検証し、変換工程でロット単位に資産変換し段階的に全資産を移行します。

	対象資産特定※	PoCサービス	コード変換サービス		機能テストサービス	テスト
		PoC	設計	変換	現新比較テスト	(IT/ST/OT)
標準期間	4～10週間	2～3か月	4～6か月	プロジェクトに依存	プロジェクトに依存	プロジェクトに依存
実施概要	• 現行資産における使用/未使用、不足/重複資産の特定 • ホスト利用機能の特定 • 移行対象資産の確定	• 特定機能の**変換及び動作検証**(対象資産特定結果を踏まえたPoC対象の調整) • サービスの**適用可否判断**	• アプリ移行概要設計、処理方式設計 • **サービス未対応のホスト技術要素のツールエンハンス**を実施 • プロトタイプによる**技術要素検証**	• 資産を**ロット分割**しロット毎にサービスによる**変換**を実施	• **変換作業完了後**、現新比較テストを実施	• 変換作業完了後、結合テスト(IT)、総合テスト(ST)、運用テスト(OT)の支援を実施 • テストによるQAや障害対応を実施

※ 対象資産特定フェーズは、モダナイゼーション関連サービス**「Fujitsu 資産分析・可視化サービス」**をご提案いたします。

凡例： リライトサービス提供範囲 / 他サービス、SI対応範囲

まずは対象資産特定やPoCでリライトサービスの実現性をご評価ください

Fujitsu PROGRESSION リライトサービス【UNIX/Linux】

- アプリケーションのビジネスロジックには手を付けずに、最も重要なCOBOL資産をJava言語へと変換します。
- アプリケーション以外（画面や帳票など）はお客様環境に応じた個別対応でのモダナイゼーション実施となります。

2026年度提供開始

	移行元（UNIX/Linux）	移行先（新環境）	
画面	HTML	React/HTML/CSS	
アプリケーション	COBOL	Java	PROGRESSIONにて対応
帳票	帳票関連	ListCreator 等	
ジョブコントロール	Shell	Shell	
データ管理	既存RDB	SQL Server/FEP/Oracle	
OS	Solaris/Linux	Linux	
ハードウェア	オンプレミス	Azure/オンプレミス	他サービスにて最適性能をご提示可能 （インフラアセスメントサービス等）

サービス適用プロセス概要 【UNIX/Linux】

1. 対象資産特定・PoCにて移行対象資産を確定し、本サービスの適用可否を判断します。
2. 後続の設計工程で技術要素を検証し、変換工程で段階的に全資産を移行します。
3. 独自のフレームワーク活用による肥大化防止＋生成AI活用で変換を最適化します。
4. テストケース生成にAIを活用し、大幅なコスト減を実現しています。
5. 全工程で富士通のレガシーモダナイゼーションの専門家「モダナイゼーションマイスター」がプロジェクトを支援します。

	対象資産特定※	PoCサービス	コード変換サービス		現新比較テスト	テスト (IT/ST/OT)	移行後運用・改修
		PoC	設計	変換			
標準期間	4～10週間	2～3か月	4～6か月	プロジェクトに依存	プロジェクトに依存	プロジェクトに依存	
実施概要	• 現行資産における使用/未使用、不足/重複資産の特定 • 移行対象資産の確定 分析に生成AI活用	• 特定機能の**変換及び動作検証**(対象資産特定結果を踏まえたPoC対象の調整) • サービスの**適用可否判断**	• アプリ移行概要設計、処理方式設計 • **サービス未対応のホスト技術要素のツールチューニング**を実施 • プロトタイプによる**技術要素検証**	• 資産を**ロット分割**しロット毎にサービスによる**変換**を実施 継続的エンハンスに生成AI活用	• **変換作業完了後**、現新比較テストを実施 テストケース生成に生成AI活用	• 変換作業完了後、結合テスト(IT)、総合テスト(ST)、運用テスト(OT)の支援を実施 • テストによるQAや障害対応を実施	生成AI活用可
モダナイゼーションマイスターの参加							

※ 対象資産特定フェーズは、モダナイゼーション関連サービス「**Fujitsu 資産分析・可視化サービス**」をご提案いたします。

凡例： リライトサービス提供範囲 / 他サービス、SI対応範囲 / 移行後の個別ご相談範囲

Fujitsu PROGRESSION リライトサービス移行事例

- 20年以上に渡り、50社以上の海外メインフレームを中心に移行に成功しています。
- 国内展開後（2024年5月以降)、メインフレーム・UNIX機での移行に向けた変換検証を実施しています。

国内

メインフレーム：7件

製造業(GS)	3件
エネルギー系(GS)	1件
公益法人系(PQ)	2件
公益法人系(IBM COBOL)	1件

UNIX：４社

キャリア系(Solaris)	1件
エネルギー系(Solaris)	1件
流通(Solaris)	1件
金融(Solaris)	1件

※１件移行実施

その他変換検証実績

海外

業種	実施時期	言語			移行先
		移行元	移行先	規模(コード行数)	
公共	2021 - 2023	COBOL	C#	2.0M step	オンプレミス / Azure
政府	2021	COBOL	Java	0.5M step	Azure
公共	2016 - 2020	COBOL	Visual Basic .NET	12.0M step	オンプレミス
学校	2018 - 2020	COBOL	C#	1.4M step	オンプレミス
金融(保険)	2018 - 2019	COBOL	C#	2.0M step	オンプレミス
金融(銀行)	2016 - 2018	COBOL	C#	1.6M step	オンプレミス
公共	2015 - 2016	COBOL	C#	1.8M step	オンプレミス
金融(保険)	2013 - 2015	COBOL	Java	1.5M step	オンプレミス
公共	2009 - 2011	COBOL	C#	1.5M step	オンプレミス

Modernization Case Study

メインフレーム リライトPoC

富士通メインフレーム　リライト　資産分析

お客様事例　機械製造業C社

補給部品管理の業務ノウハウを維持しつつシステムをオープン化する構想を検証

現行の資産分析を実施し、実際にリライトによる変換・動作・性能検証を実施するとともに、現行業務全体に関する非機能要件を含めた本格移行のための全体指針・計画を策定

課題

- 資産が膨大で、どこまで移行すべきか判別がつかない
- リライトを推進したいが、動作要件を満たせるか不安
- オープン化にあたり、どのようなシステム構成を組めば性能が担保できるのかわからない

取り組み

- 資産のスリム化に向けて、稼働状況調査を実施
- COBOLからJavaへのリライトPoCで変換品質を検証
- メインフレームとオープンの両方に精通する技術者が参画し、性能課題を見越した最適な移行パスを検討

効果

- 資産全体の8割以上が未使用であることを確認
- 自動変換率97%、機能実現性100%、現新比較テストでの障害0件と、PoC段階から高い変換品質を確認
- 性能担保の考え方を明確化し、移行パス案を具体化

適用ソリューション

Fujitsu モダナイゼーション リライトサービス Fujitsu PROGRESSION
モダナイゼーションマイスター アドバイザリーサービス

関連サービス・製品

PRIMERGY
ETERNUS

関連サイト

Fujitsu.com > モダナイズ：未来を描き、システムを最適化し、AIで加速する - DX実現へのロードマップ -

他社メインフレーム　リライト

お客様事例　公企業Ａ社

Fujitsu PROGRESSIONによる 他社メインフレームのリライト実現性実証

海外で多数の他社メインフレーム対応実績をもつFujitsu PROGRESSIONを国内に適用
3ヶ月の短期間で有効性を評価し、維持コストの高い他社機のモダナイゼーションを後押し

課題

- メインフレーム維持コストの高騰、オープン化に向けたリソース（コスト・人材）の最適化
- モダナイゼーション計画の具体化に向けた、技術課題の洗い出し・対応方針の明確化

取り組み

- COBOLソースを1行ずつJavaにリライトでき、他社機実績豊富なFujitsu PROGRESSIONでPoCを実施
- 経験豊富なマイスター※が参画。事前の資産分析・簡易変換でシステム特性を把握し、評価軸を明確化

効果

- 海外他社・国内自社の対応ノウハウを基に、450KstepのCOBOL資産Java化PoCを、3ヶ月の短期間で完了
- 大規模システムの基盤変更を前提とした技術的リスク・課題を抽出し、今後の対応指針案を提示

※モダナイゼーションマイスター：レガシーシステムのオープン化ノウハウを有する富士通の選任部隊

適用ソリューション
Fujitsu モダナイゼーションリライトサービス Fujitsu PROGRESSION

関連サービス・製品
モダナイゼーションマイスター アドバイザリーサービス
Fujitsu 資産分析・可視化サービス

関連サイト
Fujitsu.com > モダナイズ：未来を描き、システムを最適化し、AIで加速する - DX実現へのロードマップ -

無償トライアル/お試し変換

- お客様のCOBOL資産の一部を実際にJavaに変換＆評価することで、検討初期の段階でJava化の効果を体感いただけます。

こんなお悩みをお持ちのお客様へ

- モダナイゼーションはしたいが移行方式の決め手がない
- Javaへの変換のイメージが湧かない

変換前（COBOL）

```
001002     IF W状態コード１              =     '96'   OR   '97'
001003       THEN
001004         IF ＷＳＹＮＡＤ数          <     3
001005           THEN
001006* １０ｍｓｅｃ後、再試行
001007             CALL 'SUBR0220'
001008             COMPUTE ＷＳＹＮＡＤ数 = ＷＳＹＮＡＤ数 + (1)
001009             GO TO ダミーＲＥＡＤ
001010*
001011           ELSE
001012* 障害通知へ
001013             IF Ｓ部０００パスオンＷＳ    = SPACE
001014               THEN
001015                 CALL 'JXALANTE' USING ＳＱ−処理種別格納域名  ＳＱ−通知域名  ＳＱ−通知レベル域名 ＳＱ−
001016                 MOVE ＳＱ−プロシジャ名 TO Ｓ部０００プロシジャ名
001017                 MOVE 'FS' TO Ｌ部０２１依頼名
001018                 MOVE W処理コード TO Ｓ部０００処理コード
001019                 CALL 'SUBR0350' USING Ｌ部０２１画面遷移制御パラメータ  Ｓ部ＳＰＡ領域 Ｃ部０１０障害通知
001020             END-IF
```

変換後（Java）

```
if (local.W状態コード１.getValue().compareTo("96") == 0 ||
        local.W状態コード１.getValue().compareTo("97") == 0) {
    if (local.ＷｓｙｎａｄＷ数.getValue().compareTo(BigDecimal.valueOf(3)) < 0) {
        //* １０ｍｓｅｃ後、再試行
        SUBR0220.Start();
        local.ＷｓｙｎａｄＷ数.setValue(local.ＷｓｙｎａｄＷ数.getValue().add(BigDecimal.valueOf(1)));
        jumpTo("Pダミ_ｒｅａｄ");
    }
    //*
    else {

        //* 障害通知へ
        if (local.Ｓ部０００パスオンｗｓ.isSpaces()) {
            // [ProgressionCOE]: Cannot locate component 'JXALANTE', which is being called inside this progra
            JXALANTE.Start(local.Ｓｑ_処理種別格納域名, local.Ｓｑ_通知域名, local.Ｓｑ_通知レベル域名, local.Ｓｑ
            local.Ｓ部０００プロシジャ名.setValue(local.Ｓｑ_プロシジャ名.getValue());
            local.Ｌ部０２１依頼名.setValue("FS");
            local.Ｓ部０００処理コ_ド.setValue(local.W処理コ_ド.getValue());
            SUBR0350.Start(local.Ｌ部０２１画面遷移制御パラメ_タ, local.Ｓ部ｓｐａ領域, local.Ｃ部０１０障害通知メッ
        }
```

変換後のソースコード品質を体感

最短２週間でのアウトプットを提示※

※3~5本のCOBOLソース(5Kstep程度)であること、過不足なく資産をご提供いただけることが前提となります。

Fujitsu PROGRESSION
リホストサービス概要

①豊富な実績による安心確実な移行が可能

FUJITSU

特 長　既存資産を活かす安全移行

- **お客様の既存資産（レガシーシステム）を最大限に活用**
  オープン環境に対応したNetCOBOLに変換することで業務継続が可能となります
- **安全なモダナイゼーションを実施可能**
  多数プロジェクトで磨き上げられた信頼性の高い移行ツールで、お客様の重要なシステムを確実にモダナイズします。

優位性　他社を圧倒・凌駕する実績と変換力

- **豊富な実績と専門的な知見により国内90社以上の対応実績**
  COBOL特有の構造やお客様運用を理解した変換設計により、実行可能性を重視
- **富士通の豊富なモダナイゼーションによる実践知と実行力**
  多岐にわたる業種・業務における富士通のCOBOL変換実績とノウハウに基づき、お客様のビジネス要件に最適な変換を実現

②低コスト・短期間でのシステム刷新

FUJITSU

特 長　COBOL資産を継承し進化させる高互換変換技術

- **COBOL資産の互換性と将来性の両立**

  COBOL言語仕様の互換性と改定仕様への対応を行っている、NetCOBOLへの資産変換

- **COBOL以外の既存資産のオープン環境向け改修**

  JCLやDB・帳票の定義体といったCOBOL以外の既存資産を、Linux/WindowsやRDBといったオープン環境に適した資産形式に変換

- **維持・継承可能なCOBOLへの資産変換**

  国際規格非準拠のCOBOL資産を、国際規格に準拠したNetCOBOL対応に変換

優位性　独自フレームワークで既存COBOL資産をそのままの変換を実現

- **既存資産を活かし、コストと期間を圧縮**

  既存COBOL資産をそのまま新環境へ移行するため、新規開発や大規模改修に比べて、移行コストと開発期間を大幅に削減可能です。

- **ビジネスロジック変更なし、安心・確実な移行**

  長年のビジネスを支えるCOBOL資産や周辺環境は、ビジネスロジックを変更することなく、高い品質でオープン環境へ移行。

- **将来も安心、持続可能なシステム基盤**

  標準化されたNetCOBOLへ変換することで、将来の運用・保守が容易になります。技術継承リスクも低減し、長期的なシステム活用を可能にします。

Fujitsu PROGRESSION リホストサービス 【MF】

- メインフレーム用のCOBOL資産を、オープン環境で動かせるCOBOL資産へと変換いたします。
  ビジネスロジックには変更を加えずにシステムを移行することができます。
- アプリケーション以外のジョブコントロール、データ管理、帳票部分の各種定義体も、オープン環境対応に変換します。

※画面部分とトランザクション処理部分の変換対応は後報

リホストサービスを活用したモダナイゼーションの流れ 【MF】

- 確定された移行対象資産に対して、非互換調査を実施し、本サービスの変換可否を判断します。
- 後続の工程で資産の変換設計後、変換作業工程でロット単位に資産変換し段階的に全資産を移行します。

	対象資産特定※	リホストサービス			現新比較テスト	テスト (IT/ST/OT)
		非互換調査	変換設計	変換作業		
標準期間	プロジェクトに依存	2か月～	3か月～	プロジェクトに依存	プロジェクトに依存	プロジェクトに依存
実施概要	• 現行資産における使用/未使用、不足/重複資産の特定 • ホスト利用機能の特定 • 移行対象資産の確定	• 確定された**移行対象資産**の新環境との非互換を調査し非互換機能の対処方式を検討 • 調査の結果報告からサービスによる**変換可否判断**	• プロトタイプによる**技術要素検証** • **変換仕様**として非互換機能の具体的な修正方法を策定	• **非互換要素のツールエンハンス**を実施資産を**ロット分割**し、ロット毎にサービスによる**変換**を実施	• 変換作業完了後、現新比較テストの支援を実施 • テストによるQAや障害対応を実施	• 結合テスト(IT)、総合テスト(ST)、運用テスト(OT)の支援を実施 • テストによるQAや障害対応を実施

※ 対象資産特定フェーズは、モダナイゼーション関連サービス**「Fujitsu 資産分析・可視化サービス」**をご提案いたします。

凡例：
- リホストサービス提供範囲
- 他サービス、SI対応範囲

リホストサービスによる移行事例

FUJITSU

- 20年以上に渡り、90社以上の国内メインフレームを中心に移行に成功しています。
- 金融・公共、製造・流通などの分野を含む様々な業種での移行実績があります。

業種	実施時期	言語			移行先
		移行元	移行先	規模(コード行数)	
製造	2003 - 2004	COBOL	NetCOBOL	1.7M step	オンプレミス
公共	2005 - 2007	YPS/COBOL	NetCOBOL	2.0M step	オンプレミス
金融	2005 - 2008	COBOL	NetCOBOL	1.2M step	オンプレミス
金融	2008 - 2010	COBOL	NetCOBOL	2.2M step	オンプレミス
流通	2012 - 2014	COBOL	NetCOBOL	16.0M step	オンプレミス
製造	2011 - 2013	COBOL	NetCOBOL	7.0M step	プライベートクラウド
公共	2014 - 2016	YPS/COBOL	YPS/COBOL	1.8M step	オンプレミス
金融	2011 - 2013	COBOL	NetCOBOL	1.5M step	プライベートクラウド
公共	2015 - 2017	COBOL	NetCOBOL	2.5M step	オンプレミス

富士通メインフレーム　リホスト　マイスター

お客様事例　金融業某社

業務の根幹を支えるバッチ処理を安全・確実に継承

大規模なバッチ処理のクラウド化に向け、豊富なオープン化ノウハウに基づく移行ツールと、レガシー領域の高度な知見を統合し、検証計画を立案

課題

- 大規模システムで、時間制約の厳しい夜間バッチが多数存在しており、移行による性能問題の予測が困難
- バッチ処理の廃止やリライトは費用対効果が見合わない

取り組み

- メインフレームとオープンのアーキテクチャの差異をふまえ、限られた規模での最適なバッチ性能検証計画を立案
- DBアクセスツールを用いたリホストで、変更影響を最小化

効果

- 検証環境の適切なサイジングと、規模増大を想定した検証項目の抽出により、稼働後の性能問題抑止を見込む
- モダナイズに伴うアプリケーションの修正量を50%削減

適用ソリューション

モダナイゼーションマイスター アドバイザリーサービス

Fujitsu モダナイゼーション リホストサービス Fujitsu PROGRESSION

関連サービス・製品

FUJITSU Hybrid IT Service for AWS

関連サイト

レガシーシステム脱却だけではない！攻めのモダナイゼーション戦略 - Fujitsu.com

Fujitsu PROGRESSION
生成AIの取り組み

Fujitsu PROGRESSION × 生成AI

Fujitsu PROGRESSIONと生成AIを組み合わせ、サービス対応を実施中

1 変換率の向上

変換ツールのエンハンスに生成AIを活用し、お客様資産に応じた変換を実現

2 テストケースの自動生成

生成AIを活用し、未実行のテストケースを作成し、効率的かつ高品質なテストを実行

3 リファクタリング

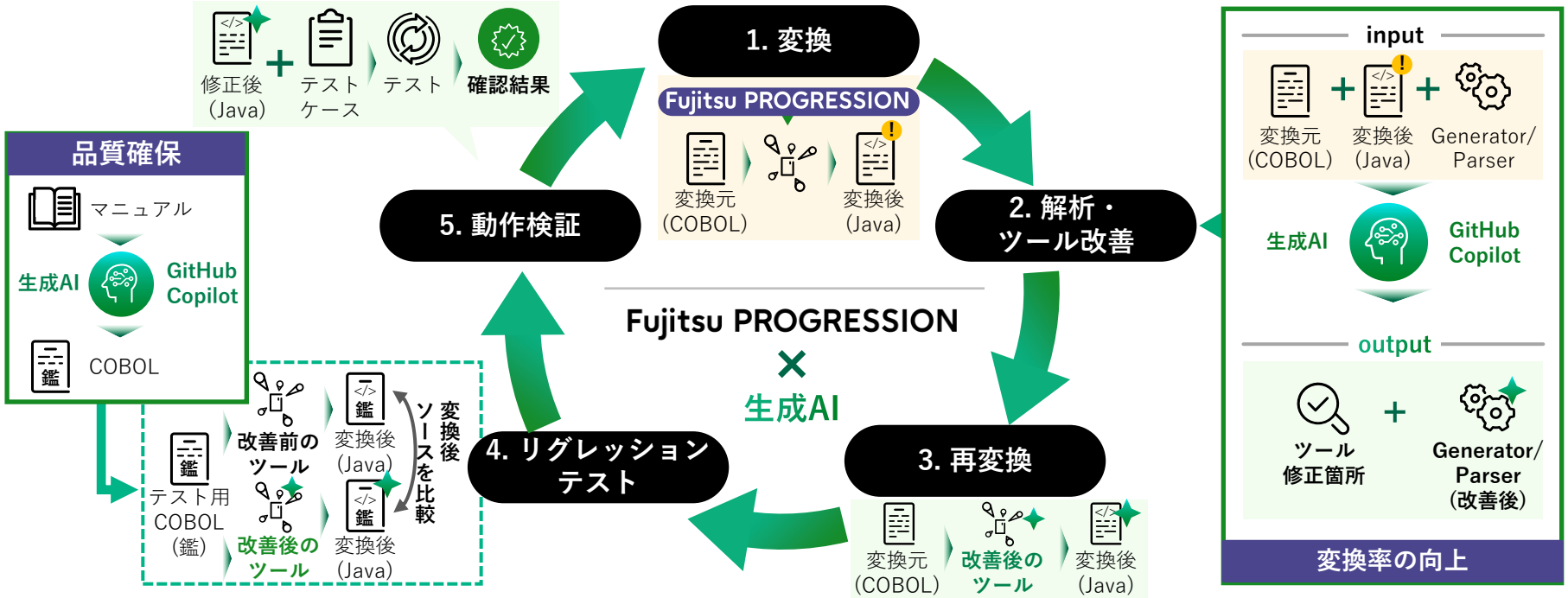
変換後ソースのリファクタリングに生成AIを活用し、お客様システムに適した改善検討を実施

生成AI活用によるQCD最適化 変換率の向上

AI

FUJITSU

PROGRESSIONは　エンハンスに生成AIを活用し、変換率の向上・品質を確保して継続的なツール改善を実践しております。

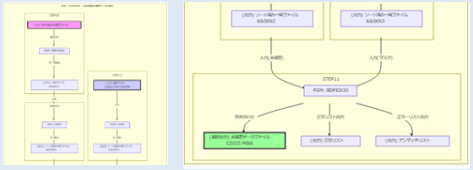


生成AIを活用したテストケースの自動生成

AI

FUJITSU

- 生成AIを活用して、テストケースを生成し、効率的なテストを実現します。
- プログラム解析・入出力ファイルの概要を抽出・インプット条件の抽出により、現行データを突合することで不足データを生成します。

	インプット		アウトプット
(1) プログラムの解析	プロンプト(指示文) JCL,COBOLプログラムを解析し、プログラム間の関係をマーメイド記法で出力してください。 ＋ ソースコード(JCL,COBOL)	生成AI	プログラムの解析結果（マーメイド記法） 
(2) 入力ファイル・処理概要の抽出	プロンプト(指示文) COBOLプログラムの実行に必要な入力ファイルを表形式で出力してください。また、実行時の処理概要を表に追記してください。 ＋ (1)のアウトプット	生成AI	入力ファイル構造 条件文の抽出結果 レコード: IFILE-REL-KEY = 0 / 管理レコード
(3) インプット条件の抽出	プロンプト(指示文) 入力ファイル・処理条件の抽出結果を元に、条件文を網羅するために必要なインプット条件を出力してください。 ＋ (2)のアウトプット	生成AI	インプット条件の抽出結果

No	データ項目	部分参照	PIC属性	設定値	REDEFINE条件	データ形式
1	IFILE-HIZUKE		9(6)			YYMMDD
2	IFILE-HIZUKE	3:4				MMDD
3	IFILE-TOTAL		S9(3)			

	DB		ファイル	
	項目１	項目２	項目A	項目B
条件１	≧1	≧3	“01”	“02”
条件２	<1	<3	“01”	“02”
条件３	＝項目A	＝項目B	＝項目１	＝項目２
条件４	＞項目A	＝項目B	＜項目１	＝項目２

現行データと突合 不足データを補完

リファクタリング

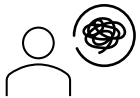
リライト後のJavaコードにおける”JABOL”要素を、生成AIによってリファクタリング

リライト後

```
protected void Main() throws Exception {
    // メイン処理（手続き型の順次実行）
    初期処理();
    while (!local.W終了フラグ.getValue().equals("END")) {
        主処理();
    }
}

private void 初期処理() throws Exception {
    initializeVariable(local.W顧客データ);
    local.W処理カウント.setValue(BigDecimal.ZERO);
    local.Wエラーカウント.setValue(BigDecimal.ZERO);
    getDbms().ready();
    顧客データ取得();
}

private void 主処理() throws Exception {
    if (local.W顧客コード.getValue().compareTo("00000") > 0) {
        データ検証();
        if (local.Wエラーフラグ.isZeroes()) {
            顧客データ更新();
            local.W処理カウント.setValue(
                local.W処理カウント.getValue().add(BigDecimal.ONE));
        } else {
            local.Wエラーカウント.setValue(
                local.Wエラーカウント.getValue().add(BigDecimal.ONE));
        }
    }
    次データ取得();
}
```



COBOL由来の構造維持

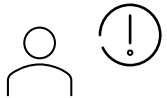
生成AI

リファクタリング後

```
public void Main() {
    // オブジェクト指向型の処理（ストリーム処理）
    初期処理();
    repository.findAll()
        .filter(data -> data.W顧客コード.compareTo("00000") > 0)
        .forEach(this::主処理);
}

private void 初期処理() {
    W処理カウント = 0;
    Wエラーカウント = 0;
    repository.open();
}

private void 主処理(CustomerData W顧客データ) {
    try {
        データ検証(W顧客データ);
        if (W顧客データ.Wエラーフラグ == 0) {
            顧客データ更新(W顧客データ);
            W処理カウント++;
        } else {
            Wエラーカウント++;
        }
    } catch (Exception e) {
        Wエラーカウント++;
    }
}
```



標準的なJava

Java技術者にとって保守・拡張しやすい標準的なJavaに進化

お問合せ先

モダナイゼーションに関するご相談・お問合せは
こちらにお願い致します

モダナイゼーションイベント事務局
・e-mail：contact-fjjp-moder@cs.jp.fujitsu.com

Thank you