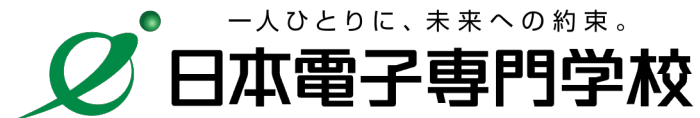


令和7年度 文部科学省委託  
専修学校による地域産業中核的人材養成事業  
(人口減少地域の職業人材を確保するための専修学校振興プログラム)

# 人口減少地域を支えるAI補完型ITエンジニア養成プログラムの開発

## 第3回事業推進委員会 検討資料

令和8年1月



# 事業推進——各年度取組

	令和7年度	令和8年度	令和9年度
趣旨	教育プログラムの基礎設計と初期教材開発	教育プログラムの中核教材拡充と教育基盤構築	教育プログラムの応用教材完成・PBL実証とモデル統合
調査	<ul style="list-style-type: none"> <li>① IT企業生成AI活用実態調査</li> <li>② IT“学び直し”ニーズ実態調査</li> <li>③ 生成AIコーディングアシスタント事例調査</li> <li>④ AI活用エンジニアリング教育事例調査</li> </ul>	<ul style="list-style-type: none"> <li>⑤ 地域AI/IT人材需給実態調査</li> <li>⑥ 地方企業DevOps実装実態調査</li> <li>⑦ DevOps/AIOps教育事例調査</li> <li>⑧ プロンプトエンジニアリング教材事例調査</li> </ul>	<ul style="list-style-type: none"> <li>⑨ 生成AI活用企業運用課題実態調査</li> <li>⑩ 地域IT人材キャリア意識調査</li> <li>⑪ DevOps人材定着施策事例調査</li> <li>⑫ AIセキュリティ教育事例調査</li> </ul>
開発	<ul style="list-style-type: none"> <li>01 AI駆動ITスキル指標設計</li> <li>02 GPT-TA評価システム設計</li> <li>03 教育プログラム全体カリキュラム開発</li> <li>04 教育プログラムシラバス作成</li> <li>05 教育プログラム講義動画制作</li> <li>06 教育プログラム講義教材作成</li> <li>07 教育プログラム演習課題作成</li> <li>08 教育プログラム自習問題作成 ※ 対象科目：S1～S2前 (12コマ)</li> <li>09 教育プログラム基盤 要件定義</li> </ul>	<ul style="list-style-type: none"> <li>10 AI駆動ITスキル指標開発</li> <li>11 GPT-TA評価システムβ実装</li> <li>12 教育プログラムシラバス作成</li> <li>13 教育プログラム講義動画制作</li> <li>14 教育プログラム講義教材作成</li> <li>15 教育プログラム演習課題作成</li> <li>16 教育プログラム自習問題作成 ※ 対象科目：S2後～S4 (23コマ)</li> <li>17 教育プログラムPBL設計書作成</li> <li>18 教育プログラム基盤β構築運用</li> </ul>	<ul style="list-style-type: none"> <li>19 BASIS×GPT-TA構築運用</li> <li>20 教育プログラムシラバス作成</li> <li>21 教育プログラム講義動画制作</li> <li>22 教育プログラム講義教材作成</li> <li>23 教育プログラム演習課題作成</li> <li>24 教育プログラム自習問題作成 ※ 対象科目：S5～S6 (13コマ)</li> <li>25 教育プログラムPBL教材作成</li> <li>26 教育プログラム基盤構築運用</li> <li>27 企業DX診断課題バンク整備</li> <li>28 修了後メンタリング体制整備</li> </ul>
実証	—	<ul style="list-style-type: none"> <li>◎ 先行実証講座 (2週間) 実施 ※ 対象科目：S1～S2前</li> </ul>	<ul style="list-style-type: none"> <li>◎ 統合実証講座 (6週間) 実施 ※ 対象科目：S1～S4+ミニPBL</li> </ul>

# 事業実施——本年度取組 (9～2月)

タスク		9月			10月			11月			12月			1月			2月			3月			
		上	中	下	上	中	下	上	中	下	上	中	下	上	中	下	上	中	下	上			
会議	○ 事業推進委員会・分科会 開催						● 10/27						● 12/15						● 01/28				3月1日 文部科学省に提出
	○ 成果報告会 開催																		○				
調査	① IT企業生成AI活用実態調査									→													文部科学省と契約締結(II事業開始)
	② IT“学び直し”ニーズ実態調査									→													
	③ 生成AIコーディングアシスタント事例調査						→																
	④ AI活用エンジニアリング教育事例調査						→																
開発	01 AI駆動ITスキル指標設計									→													取り纏め & 事業報告書作成
	02 GPT-TA評価システム設計						→																
	03 教育プログラム全体カリキュラム開発			→																			
	04 教育プログラムシラバス作成									→			→			→							
	05 教育プログラム講義動画制作												→			→							
	06 教育プログラム講義教材作成												→			→							
	07 教育プログラム演習課題作成															→			→				
	08 教育プログラム自習問題作成															→			→				
	09 教育プログラム基盤 要件定義															→			→				

01/28時点

# 事業概要——人口減少地域を支えるAI補完型ITエンジニア養成プログラムの開発

## 事業目的

生成AIを駆使して情報システム開発の速度と質を飛躍的に向上させた次世代型IT人材=AI補完型ITエンジニアを人口減少地域において持続的に養成し、地域産業のDXを自走させる基盤を構築する。

## 事業内容

- 専修学校が中核になってAI駆動ITスキル指標を策定し、これを“礎”にした教育プログラムを開発する。
- その教育プログラムの実装・運用を支える教育環境・評価枠組・普及体制等を束ねることを狙い、実務即応・地域定着・全国展開を同時に可能にする教育モデルを構築する。

### ① 「AI補完型ITエンジニア養成プログラム」の開発

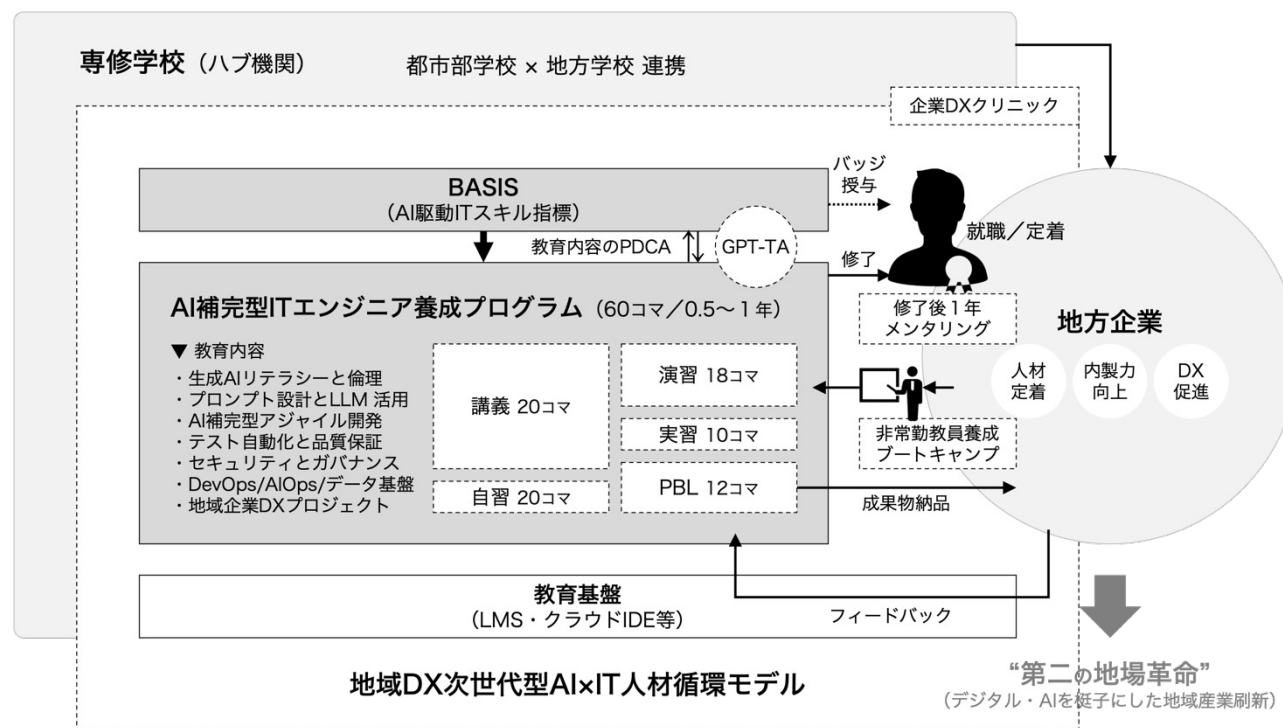
AI駆動ITスキル指標に準拠した60コマ程度の生成AI×DevOpsカリキュラムとGPTティーチングアシスタント評価システムを運用し、0.5～1年間で「要件定義→実装→運用自動化」を自走できるエンジニアを養成する。

### ② 地域制約を超える“ハイブリッド教育基盤”の構築

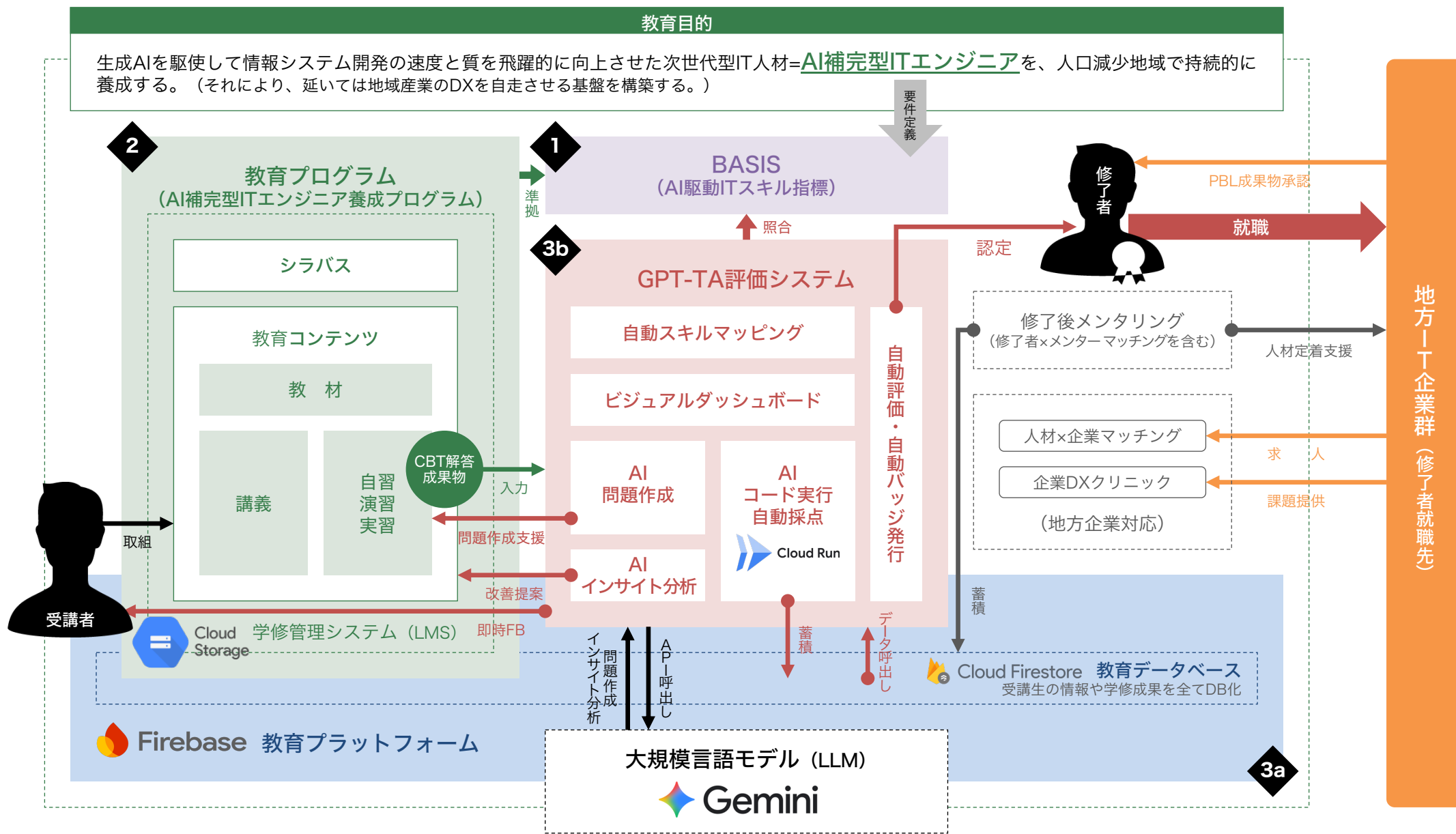
遠隔講義・クラウド統合開発環境・非常勤エンジニア教員養成ブートキャンプを組み込むことで、地理的制約や教員不足を克服し、人口減少地域でも同質の教育を実現する。

### ③ 教育-評価-産学連携の“循環エコシステム”の整備

AI駆動ITスキル指標に基づくバッジ授与によるスキルレベルの可視化、企業DX達成度評価と教育プログラムPBLの即時マッチング、修了後1年メンタリングを組み合わせ、修了者のキャリア定着率85%、企業の内製DX案件化率70%を目標にした支援環境を構築する。



# 教育システム——全体設計



# BASIS (AI駆動ITスキル指標) —— AI補完型ITエンジニアの人材要件

## ● 次世代型IT人材

### ▼ 「事業計画書」では……

- ✓ 生成AIを武器にした次世代型IT人材
- ✓ AIコーディングアシスタントを駆使する人材
- ✓ AIが出力した成果物を精査して業務案件に落とし込めるエンジニア
- ✓ AIを有効に活用しながら情報システムを高速かつ高品質に内製できる人材
- ✓ 生成AIを駆使して情報システム開発の速度と質を飛躍的に向上させた次世代型IT人材
- ✓ 生成AIを駆使して情報システム開発のプロセスを高速化・高品質化できる「AI補完型ITエンジニア」
- ✓ 生成AIの力を実務に応用し、情報システムの要件定義から運用改善まで継続的に遂行・主導できる次世代型IT人材
- ✓ 「要件定義→実装→運用自動化」を自走できるエンジニア
- ✓ 生成AIとDevOpsの一体的活用によって要件定義から運用自動化まで自律的に遂行できる人材
- ✓ 地方で育って地方で働く次世代型IT人材
- ✓ 地方企業の内製力を底上げするDXの中核を担う人材
- ✓ 地方企業の実課題をAI駆動ITスキルによって解決し、継続的改善サイクルを組織内で主導できる人材

## ▼ AI補完型ITエンジニア

生成AIを安全かつ適切に扱い、倫理的に運用できる力

モデルの限界・幻覚・バイアスを理解し、国際規格や組織ポリシーに準拠して生成物の事実性・一貫性を評価して扱える。

業務要件を設計・プロンプト・LLM連携に変換できる力

自然言語の要件をユーザーストーリー/受入基準と高精度プロンプトに落とし込み、RAG・API・マルチモーダル連携まで設計して実装可能にできる。

生成AI補完で短サイクルに実装し、テスト自動化で品質を保証できる力

Copilot等で実装を加速しつつ、静的解析・TDD/BDD・品質ゲートで“速度×品質”のトレードオフを統制できる。

運用をセキュアに自動化し、本番移行・引継ぎまで遂行できる力

生成AI特有の脅威と法令に備え、CI/CD・IaC・Observability・AIOpsで運用を自動化し、“UAT→本番移行→運用手順化”を小規模チームで完遂できる。

## ▼ AI駆動ITスキル指標 (BASIS : Benchmark for AI-Supported IT Skills)

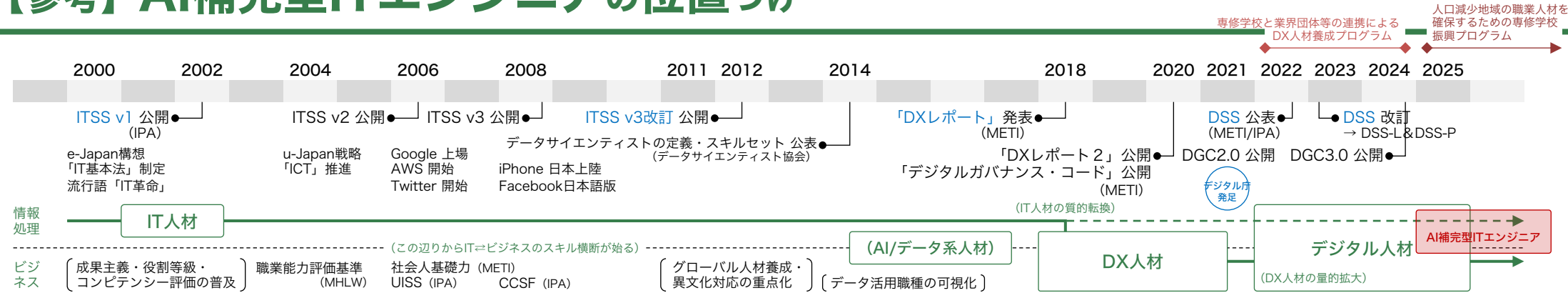
本事業で養成を目指すAI補完型ITエンジニアのスキル要件を定義し、その修得度を評価するための指標。

この人材が0.5~1年で(要件定義→実装→運用自動化)を自走できるよう、汎用ITスキルをAIで高速化・高精度化する観点で設計。“AIはツール、人は評価と最適化”との役割意識を前提にしてスキルと判断力を同時に測定。

大分類 (ドメイン)	中分類 (サブドメイン)
A 生成AIリテラシー/ 倫理・規格	A1 モデル原理・限界・幻覚・バイアスの理解
	A2 倫理・国際規格・ガバナンスの適用
	A3 組織内AIポリシーの理解・遵守
	A4 生成品質の基本評価観点 (事実性・一貫性・再現性)
B プロンプト設計/ LLM活用	B1 プロンプト設計パターン (役割付与・CoT・分割統治)
	B2 出力評価と再プロンプトによる改善ループ
	B3 LLM API/RAGの実装
	B4 コスト/レイテンシ最適化とスロットリング設計
C 要件定義/設計変換	C1 ビジネス要件のユーザーストーリー/受入基準への変換
	C2 非機能要件 (性能・保守・運用) の定義
	C3 設計案の比較評価・採択 (AI提案の選択眼)
	C4 バックログ編成とスプリント計画
D AI補完型アジャイル実装	D1 生成補完によるCRUDの実装 (Copilot等の安全活用)
	D2 レビュー・リファクタの人間主導判断
	D3 API/DB連携とエラーハンドリング設計
	D4 チーム開発の運用 (Git・PR・課題トラッキング)
E テスト自動化/品質保証	E1 テスト戦略とTDD/BDD
	E2 LLMを用いたテスト生成とフェイルファスト
	E3 静的解析・SCA・SAST/DASTの統合
	E4 品質メトリクス統合と品質ゲート運用
F セキュリティ/ 法規・ガバナンス	F1 生成AI特有脅威 (プロンプトインジェクション等) の対策
	F2 データ保護 (PIIを含む) と著作権・ライセンス遵守
	F3 生成物の安全性・真正性チェック (SBOM等を含む実装)
	F4 監査ログ設計と利用履歴トレーサビリティ確保
G DevOps/AIOps/ データ基盤	G1 CI/CDパイプラインの設計 (VCS~自動デプロイ)
	G2 IaC・コンテナ・Kubernetesによる環境標準化
	G3 観測可能性 (ログ・トレース・APM) の実装
	G4 AIOpsによる異常検知と改善提案の運用
H 実装移行/運用引継ぎ	H1 MVP実装とユーザー検証
	H2 UAT実施と「実装可」判定取得
	H3 本番移行計画・ドキュメント・運用手順の整備
	H4 改善スプリントでのSLO/KPIの達成

開発する教育プログラムのカリキュラムの到達目標を可視化し、GPT-TA自動採点結果をドメイン別バッジとして発行。

# 【参考】AI補完型ITエンジニアの位置づけ



## DSS-Pを構成する人材類型

### ◆ ビジネスアーキテクト

DXによるビジネス変革の目的を定義し、関係者をコーディネートして〈構想→実装→効果検証〉を一貫して推進する統合的人材。

【ロール】 ビジネスアーキテクト (新規事業開発)、ビジネスアーキテクト (既存事業の高度化)、ビジネスアーキテクト (社内業務の高度化・効率化)

### ◆ デザイナー

顧客・ユーザーとビジネスの視点を統合し、製品・サービスの方針と開発のプロセスを策定して体験や表現をデザインする人材。

【ロール】 サービスデザイナー、UX/UIデザイナー、グラフィックデザイナー

### ◆ データサイエンティスト

DXによる業務変革や新規ビジネス実現に向けてデータ収集・解析の仕組を設計・実装・運用し、データ活用を牽引する人材。

【ロール】 データビジネスストラテジスト、データサイエンスプロフェッショナル、データエンジニア

### ◆ ソフトウェアエンジニア

デジタル技術を用いた製品・サービス提供のために、システムやソフトウェアを設計・実装・運用する中核的人材。

【ロール】 フロントエンドエンジニア、バックエンドエンジニア、クラウドエンジニア/SRE、フィジカルコンピューティングエンジニア

### ◆ サイバーセキュリティ

業務を支えるデジタル環境のリスクを抑制するため、方針策定から対策の導入・運用までを担う「守り」の専門人材。

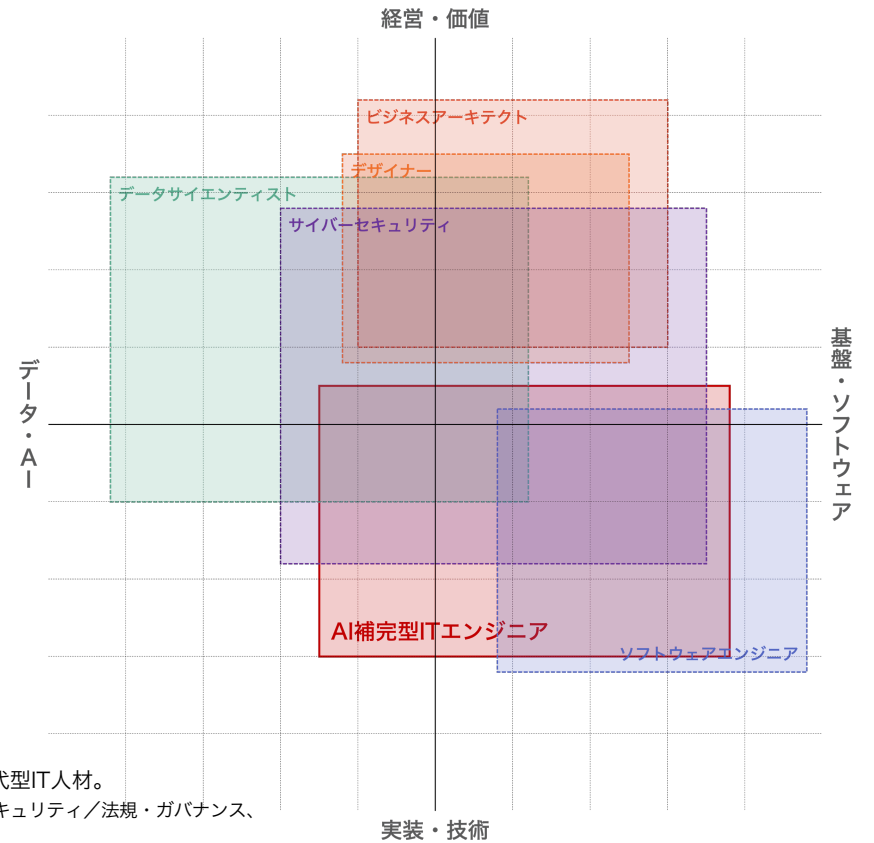
【ロール】 サイバーセキュリティマネージャー、サイバーセキュリティエンジニア

## 本事業で養成する人材

### ◆ AI補完型ITエンジニア

生成AIを駆使して情報システム開発の速度と質を飛躍的に向上させ、要件定義から運用改善まで継続的に遂行・主導できる次世代型IT人材。

【スキル】 生成AIリテラシー/倫理・規格、プロンプト設計/LLM活用、要件定義/設計変換、AI補完型アジャイル実装、テスト自動化/品質保証、セキュリティ/法規・ガバナンス、DevOps/AIOps/データ基盤、実装移行/運用引継ぎ





# BASIS × iCD/DSS/ITSS 対照評価表——ドメインA～D

BASIS		iCD			DSS			ITSS						
大分類	中分類	スキル分類	評価 適合種	対応度	優先度	カテゴリ：サブカテゴリ：スキル項目	評価 適合種	対応度	優先度	職種：スキル項目	評価 適合種	対応度	優先度	
A 生成AIリテラシー/ 倫理・規格	A1	モデル原理・限界・幻覚・バイアスの理解	(システム) ソフトウェアの基礎技術	△	D	中	テクノロジー： デジタルテクノロジー： テクノロジートレンド	△	C	中	ソフトウェア開発： 分析・要求定義	△	D	中
	A2	倫理・国際規格・ガバナンスの適用	(関連知識) 法規・基準・標準	△	C	高	セキュリティ： セキュリティマネジメント： プライバシー保護	△	C	高	ソフトウェア開発： 分析・要求定義	△	C	高
	A3	組織内AIポリシーの理解・遵守	(支援活動) ITガバナンス	c	C	高	セキュリティ： セキュリティマネジメント： セキュリティマネジメント	△	C	高	ITアーキテクト： 標準化と再利用	△	C	高
	A4	生成品質の基本評価観点 (事実性・一貫性・再現性)	(支援活動) 品質マネジメント手法	c	B	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	△	C	高	ソフトウェア開発： 開発方式設計	c	B	高
B プロンプト設計/ LLM活用	B1	プロンプト設計パターン (役割付与・CoT・分割統治)	(企画) 要求分析手法	△	C	高	テクノロジー： ソフトウェア開発： ソフトウェア設計手法	△	C	高	ITアーキテクト： 設計技法	△	C	高
	B2	出力評価と再プロンプトによる改善ループ	(支援活動) 品質マネジメント手法	c	B	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	c	B	高	ソフトウェア開発： 開発方式設計	△	C	高
	B3	LLM API/RAGの実装	(システム) Webシステムの構築技術	c	B	高	テクノロジー： ソフトウェア開発： バックエンドシステム開発	△	C	高	ITアーキテクト： アーキテックチャ設計	△	C	高
	B4	コスト/レイテンシ最適化とスロットリング設計	(非機能要件) 非機能要件(可用性・性能・拡張性)	c	B	高	テクノロジー： ソフトウェア開発： SREプロセス	△	C	高	ITアーキテクト： 設計技法	△	C	高
C 要件定義/ 設計変換	C1	ビジネス要件のユーザーストーリー/ 受入基準への変換	(企画) 要求分析手法	c	B	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	△	C	高	ソフトウェア開発： 分析・要求定義	c	B	高
	C2	非機能要件(性能・保守・運用)の定義	(企画) 非機能要件設計手法	c	B	高	テクノロジー： ソフトウェア開発： SREプロセス	△	C	高	ITアーキテクト： 設計技法	c	B	高
	C3	設計案の比較評価・採択 (AI提案の選択眼)	(実装) アーキテックチャ設計手法	△	C	高	テクノロジー： ソフトウェア開発： ソフトウェア設計手法	△	C	高	ITアーキテクト： アーキテックチャ設計	△	C	高
	C4	バックログ編成とスプリント計画	(実装) プロジェクトマネジメント手法	△	C	中	テクノロジー： ソフトウェア開発： チーム開発	△	C	中	ソフトウェア開発： 開発方式設計	△	C	中
D AI補完型 アジャイル実装	D1	生成補完によるCRUDの実装 (Copilot等の安全活用)	(システム) ソフトウェアの構築技術	c	B	高	テクノロジー： ソフトウェア開発： バックエンドシステム開発	c	B	高	ソフトウェア開発： ソフトウェアエンジニアリング	c	B	高
	D2	レビュー・リファクタの 人間主導判断	(実装) ソフトウェアエンジニアリング手法	c	B	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	c	B	高	ソフトウェア開発： ソフトウェアエンジニアリング	c	B	高
	D3	API/DB連携と エラーハンドリング設計	(システム) ソフトウェアの構築技術	c	B	高	テクノロジー： ソフトウェア開発： Webアプリケーション基本技術	c	B	高	ソフトウェア開発： ソフトウェアエンジニアリング	c	B	高
	D4	チーム開発の運用 (Git・PR・課題トラッキング)	(支援活動) 標準化・再利用手法	△	C	高	テクノロジー： ソフトウェア開発： チーム開発	c	B	高	ソフトウェア開発： ソフトウェアエンジニアリング	c	B	高

# BASIS × iCD/DSS/ITSS 対照評価表——ドメインE～H

BASIS		iCD			DSS			ITSS						
大分類	中分類	スキル分類	評価			カテゴリ：サブカテゴリ：スキル項目	評価			職種：スキル項目	評価			
			適合種	対応度	優先度		適合種	対応度	優先度		適合種	対応度	優先度	
E テスト自動化/ 品質保証	E1	テスト戦略とTDD/BDD	(支援活動) 品質マネジメント手法	c	B	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	c	B	高	ソフトウェア開発： 開発方式設計	c	B	高
	E2	LLMを用いたテスト生成と フェイルファスト	(支援活動) 品質マネジメント手法	△	C	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	△	C	高	ソフトウェア開発： ソフトウェアエンジニアリング	c	B	高
	E3	静的解析・SCA・SAST/DASTの統合	(支援活動) 品質マネジメント手法	△	C	高	セキュリティ： セキュリティ技術： セキュア設計・開発・構築	c	B	高	ITスペシャリスト： セキュリティシステム構築	c	B	高
	E4	品質メトリクス統合と品質ゲート運用	(支援活動) 品質マネジメント手法	c	B	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	△	C	高	ソフトウェア開発： 開発方式設計	△	C	高
F セキュリティ/ 法規・ガバナンス	F1	生成AI特有脅威 (プロンプトインジェクション等)の対策	(非機能要件) セキュリティの構築技術	△	C	高	セキュリティ： セキュリティ技術： セキュア設計・開発・構築	△	C	高	ITスペシャリスト： セキュリティ要素関連技術	△	C	高
	F2	データ保護 (PIIを含む) と 著作権・ライセンス遵守	(関連知識) 法規・基準・標準	△	C	高	セキュリティ： セキュリティマネジメント： プライバシー保護	△	C	高	ソフトウェア開発： 分析・要求定義	△	C	高
	F3	生成物の安全性・真正性チェック (SBOM等を含む実装)	(非機能要件) セキュリティの構築技術	△	D	高	セキュリティ： セキュリティ技術： セキュア設計・開発・構築	△	C	高	ITスペシャリスト： セキュリティシステム構築	△	C	高
	F4	監査ログ設計と 利用履歴トレーサビリティ確保	(支援活動) システム監査手法	c	B	高	セキュリティ： セキュリティマネジメント： セキュリティ体制構築・運営	△	C	高	ITサービスマネジメント： セキュリティ管理	c	B	高
G DevOps/AIOps/ データ基盤	G1	CI/CDパイプラインの設計 (VCS～自動デプロイ)	(開発) システム開発管理技術	c	B	高	テクノロジー： ソフトウェア開発： SREプロセス	c	B	高	ソフトウェア開発： ソフトウェアエンジニアリング	c	B	高
	G2	laC・コンテナ・Kubernetesによる 環境標準化	(システム) プラットフォームの構築技術	c	C	高	テクノロジー： ソフトウェア開発： クラウドインフラ活用	c	B	高	ITスペシャリスト： プラットフォームシステム構築	c	B	高
	G3	観測可能性 (ログ・トレース・APM) の実装	(保守・運用) システム保守・運用・評価	c	B	高	テクノロジー： ソフトウェア開発： SREプロセス	c	B	高	ITサービスマネジメント： ITサービスオペレーション	c	B	高
	G4	AIOpsによる異常検知と改善提案の運用	(保守・運用) システム保守・運用・評価	△	C	中	テクノロジー： ソフトウェア開発： SREプロセス	△	C	中	ITサービスマネジメント： ITサービスマネジメント業務管理	△	C	中
H 実装移行/ 運用引継ぎ	H1	MVP実装とユーザー検証	(利活用) サービスの設計・移行	c	B	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	△	C	高	ITサービスマネジメント： サービスデリバリー	△	C	高
	H2	UAT実施と「実装可」判定取得	(利活用) サービスの設計・移行	△	C	高	テクノロジー： ソフトウェア開発： ソフトウェア開発プロセス	△	C	高	ITサービスマネジメント： サービスデリバリー	△	C	高
	H3	本番移行計画・ドキュメント・ 運用手順の整備	(利活用) サービスの設計・移行	c	B	高	テクノロジー： ソフトウェア開発： SREプロセス	△	C	高	ITサービスマネジメント： サービスサポート	c	B	高
	H4	改善スプリントでのSLO/KPIの達成	(保守・運用) システム保守・運用・評価	c	B	高	テクノロジー： ソフトウェア開発： SREプロセス	c	B	高	ITスペシャリスト： システム管理要素関連技術	c	B	高

# BASISから教育プログラムへ

## ▼ AI駆動ITスキル指標 (BASIS : Benchmark for AI-Supported IT Skills) 枠組

大分類 (ドメイン)	中分類 (サブドメイン)
A 生成AIリテラシー／倫理・規格	A1 モデル原理・限界・幻覚・バイアスの理解
	A2 倫理・国際規格・ガバナンスの適用
	A3 組織内AIポリシーの理解・遵守
	A4 生成品質の基本評価観点 (事実性・一貫性・再現性)
B プロンプト設計／LLM活用	B1 プロンプト設計パターン (役割付与・CoT・分割統治)
	B2 出力評価と再プロンプトによる改善ループ
	B3 LLM API/RAGの実装
	B4 コスト/レイテンシ最適化とスロットリング設計
C 要件定義／設計変換	C1 ビジネス要件のユースケース/受入基準への変換
	C2 非機能要件 (性能・保守・運用) の定義
	C3 設計案の比較評価・採択 (AI提案の選択眼)
	C4 バックログ編成とスプリント計画
D AI補完型アジャイル実装	D1 生成補完によるCRUDの実装 (Copilot等の安全活用)
	D2 レビュー・リファクタの人間主導判断
	D3 API/DB連携とエラーハンドリング設計
	D4 チーム開発の運用 (Git・PR・課題トラッキング)
E テスト自動化／品質保証	E1 テスト戦略とTDD/BDD
	E2 LLMを用いたテスト生成とフェイルファスト
	E3 静的解析・SCA・SAST/DASTの統合
	E4 品質メトリクス統合と品質ゲート運用
F セキュリティ／法規・ガバナンス	F1 生成AI特有脅威 (プロンプトインジェクション等) の対策
	F2 データ保護 (PIIを含む) と著作権・ライセンス遵守
	F3 生成物の安全性・真正性チェック (SBOM等を含む実装)
	F4 監査ログ設計と利用履歴トレーサビリティ確保
G DevOps/AIOps／データ基盤	G1 CI/CDパイプラインの設計 (VCS~自動デプロイ)
	G2 IaC・コンテナ・Kubernetesによる環境標準化
	G3 観測可能性 (ログ・トレース・APM) の実装
	G4 AIOpsによる異常検知と改善提案の運用
H 実装移行／運用引継ぎ	H1 MVP実装とユーザー検証
	H2 UAT実施と「実装可」判定取得
	H3 本番移行計画・ドキュメント・運用手順の整備
	H4 改善スプリントでのSLO/KPIの達成

## ▼ AI補完型ITエンジニア養成プログラム：科目構成

科目名	科目目的 [主要単元]	コマ	L	E	P
S1 生成AIリテラシーとAI倫理	生成AIの基礎構造・限界・社会的影響を理解し、その利用のリスクと影響を評価しながら適切な利用行動を判断した上で、組織内AIポリシーを理解・遵守できる素養を形成する。 [生成AI概論、AI倫理とガバナンス、モデル限界・バイアス、標準化・量子化、国際AI倫理規格、等]	6	4	2	0
S2 プロンプト設計とLLM活用	LLMの内部機構とAPI実装を理解し、出力評価と再プロンプトによる改善ループを通じて高品質の出力を得るプロンプト設計とRAG (文書検索で根拠を附与して回答精度を高める手法) 構築を習得する。 [LLMアーキテクチャ、プロンプト設計手法、LLM API実装、RAG応用、マルチモーダルプロンプト、生成品質評価、等]	12	6	4	2
S3 AI補完型アジャイル開発	Copilot等の生成AIツールを組み込んだアジャイル開発手法を学び、受入基準に基づいて短サイクルで機能実装・リリースできるチーム開発力を涵養する。 [アジャイル概論、バックログ管理、Copilot CRUD、レビュー・リファクタ、リリース計画・KPI、等]	9	3	4	2
S4 テスト自動化と品質保証	Mutation TestやLLMを用いたテスト生成で品質を測定・維持する技術を学び、CIと品質ゲートを設計できる力を涵養する。 [テスト自動化戦略、カバレッジ最適化・TDD、品質メトリクス統合、品質ゲートライン設計、等]	8	3	3	2
S5 セキュリティとガバナンス	生成AI特有の脅威を理解し、安全な利用ポリシーと (データ保護および著作権・ライセンス遵守を含む) 法的コンプライアンス対応策を実践できる力を涵養する。 [生成AIセキュリティ脅威、プロンプトインジェクション対策、法規制・コンプライアンス、等]	4	2	1	1
S6 DevOps/AIOps／データ基盤	IaC・Kubernetes・AIOpsを組み合わせ、CI/CDによるデプロイと運用自動化を実装できるプラットフォームエンジニアリングを習得する。 [DevOps文化・CI/CD、GitHub Actions演習、IaC・K8s、AIOps・Observability、データパイプライン設計、等]	9	2	4	3
PBL 地域DXプロジェクト	地方企業のDX課題をチームで解決し、生成AIとDevOpsの実務適用経験を通じてBASISバッジ上位レベルを取得する。 [前期：要件定義→MVP、後期：改善スプリント→KPI達成・納品]	12	3	3	6
(コマ数) 合計		60	23	21	16

# 教育プログラム——仕様

## ▼ 教育プログラム概要

名称	AI補完型ITエンジニア養成プログラム
概要	人口減少地域でも企業DXを推進・持続させるべく、AI駆動ITスキル指標（BASIS）に基づいてAI×DevOpsの教育内容を体系化した上で、測定可能な循環型教育と地方企業課題解決PBLを融合し、AI補完型ITエンジニアを0.5～1年間で養成する専修学校発の職業教育プログラム。
ポリシー	<ul style="list-style-type: none"><li>○ 教育目標をBASISレベルに直結し、科目内容・評価をBASISに対応させる。</li><li>○ 全体の2/3をハンズオン型教育（演習・実習・PBL）にして実践型教育を実現する。</li><li>○ GPT-TAを活用したCBT解答・提出物等の採点・講評によって学修成果を評価する。</li><li>○ GPT-TA集計データの分析によって教育プログラムの質保証と継続的改善を実施する。</li></ul>
教育目標	<ul style="list-style-type: none"><li>○ 生成AIとDevOpsの一体的活用によって要件定義から運用自動化まで自律的に遂行できる能力を習得させる。</li><li>○ 地方企業の実課題をAI駆動ITスキルによって解決し、継続的改善サイクルを組織内で主導できる人材にする。</li></ul>
教育対象	<ul style="list-style-type: none"><li>○ IT専門学校（情報処理系学科）卒業生・在校生（2年生以上推奨）</li><li>○ 地方企業所属の現役エンジニア</li><li>○ リスキリング目的の非IT社会人 等</li></ul>
修了判定	<ul style="list-style-type: none"><li>○ GPT-TAによるスコア算出方法と判定ロジックに基づき、全科目で65点以上（100点中）を得点</li><li>○ BASISの全ドメインでバッジLv.1以上を取得</li><li>○ PBL成果物に対して企業が「実装可」と承認</li></ul>
学修成果（目標）	<ul style="list-style-type: none"><li>○ BASISバッジ全ドメイン取得率90%</li><li>○ PBL成果物 企業導入率70%</li><li>○ 修了後1年定着率85%</li></ul>

## ▼ 教育形態

講義（Lecture）	教員主導で理論・枠組等を体系的に解説する授業 [33.3%]
自習（Self-Study）	各講義後20分で実施するCBT（正答率80%未満は自動再受験） [授業外]
演習（Exercise）	講義内容を踏まえて教員が工程・手順を提示しながら小規模タスクを実装するラボ授業 [30.0%]
実習（Practical Training）	受講者が半自律的にサブシステムや機能を実装して講評を受ける開発実務型授業 [16.7%]
PBL（Project-Based Learning）	地方企業の実課題をチームで分析・設計・実装・検証して成果発表する課題解決型授業 [20.0%]

# 教育プログラム——シラバス【科目S1】

## ▼【科目S1】生成AIリテラシーとAI倫理（講義4コマ・演習2コマ）

生成AIの基礎構造・限界・社会的影響を理解し、その利用のリスクと影響を評価しながら適切な利用行動を判断した上で、組織内AIポリシーを理解・遵守できる素養を形成する。

コマ名	01 生成AIの基礎と安全利用の前提	02 誤りの起こり方と確かめ方	03 倫理・権利・プライバシー
コマ概要	生成AIが何を行う技術かを整理し、入力と出力の関係、プロンプトの基本形、目的の提示の仕方等を学習する。検索との違いや限界を把握し、業務利用時の注意点や避けるべき事例を共有する。安全利用のための最小限のルールを明確にする。	出力に誤りが混入する仕組みを理解し、事実でない文が尤もらしく出る現象の見分け方を学習する。信頼できる情報源の参照、複数資料の照合、言い切り表現の点検等、簡単に試せる確かめ方を習得し、過剰な自動化を避ける姿勢を確認する。	入力情報と出力結果の関係を理解し、個人情報の扱い、著作権やライセンスへの配慮、引用と帰属の基本等を学習する。業務で避けるべき入力、持出禁止の情報、共有時の注意等を具体的に整理し、迷った時に（上長等の）責任者に確認する手順を把握する。
科目内での当該コマの位置付け	科目S1全体の土台として、共通語彙と前提理解を確立するコマである。この後に学ぶ倫理・品質の問題（03・04）や演習（05・06）で扱う用語と適切利用の判断基準を先に捉え、生成AIに関する誤解を減じて以降の学習を進め易くする。	01を踏まえ、誤りの混入を前提にして安全・保守的に確かめる視点を獲得するコマである。この後に学ぶ倫理・品質の問題（03・04）や演習（05・06）の検証手順に接続するため、そのチェック観点と再確認の流れを整備する。	安全利用の基礎（01）を倫理・権利の観点で補強するコマである。この後に学ぶ組織内ポリシー（04）や権利表示演習（06）に備え、守るべき事項を最小限の条項に整理し、実務に使える言い回しと連絡の流れを整備する。
コマ形態	講義（オンデマンド視聴）	講義（オンデマンド視聴）	講義（オンデマンド視聴）
コマ内容	<ul style="list-style-type: none"> <li>○ 生成AIは何か／LLMの概観</li> <li>○ 検索と生成の違いと使い分け</li> <li>○ プロンプトの基本形と目的の提示の仕方</li> <li>○ 初期設定と安全利用の基本</li> </ul>	<ul style="list-style-type: none"> <li>○ 誤りの型の基本（混同・作り話・過度な一般化等）</li> <li>○ 確かめ方の基本（出典確認・複数照合等）</li> <li>○ 言い切り表現の点検と書き直し</li> <li>○ 自動化の限界と利用者による最終確認</li> </ul>	<ul style="list-style-type: none"> <li>○ 個人情報と機微情報の取扱い</li> <li>○ 著作権・ライセンスの基本と引用</li> <li>○ 共有時の注意と社内外の線引き</li> <li>○ 迷った時の相談と記録の残し方</li> </ul>
キーワード	生成AI LLM（大規模言語モデル） 検索 プロンプト 目的定義 出力形式 入力禁止 安全利用	誤り 幻覚 検証 出典確認 複数照合 断定回避 フォールバック 最終確認	個人情報 機微情報 入力可否 著作権 ライセンス 引用 帰属 相談
到達目標	<ul style="list-style-type: none"> <li>◇ 生成AIの出力が確率的で条件によって揺れることを理解し、検索等の決定論的処理との違いを説明できる。</li> <li>◇ 生成AIの特性と限界を踏まえ、業務等で適切利用の判断基準が必要になる理由を説明できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ 生成AIにおいて誤りや幻覚が生ずる仕組みを理解し、それがいかなる場面で問題化するかを説明できる。</li> <li>◇ 出力を無条件に信頼できない理由を理解し、確認が必要になる典型例を挙げて説明できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ 個人情報・著作権・ライセンス等の基本的な考え方を理解し、生成AI利用時に注意すべき点を説明できる。</li> <li>◇ 引用や帰属が必要になる理由を理解し、適切でない利用例を説明できる。</li> </ul>
理解度確認テスト（出題観点）	<ol style="list-style-type: none"> <li>① 生成AIと検索の差異識別</li> <li>② プロンプト必須要素の選別</li> <li>③ 目的適合プロンプト例の選択</li> <li>④ 入力禁止情報の判別</li> <li>⑤ 出力変動要因と抑制策の選択</li> </ol>	<ol style="list-style-type: none"> <li>① 幻覚（事実性欠如）例の識別</li> <li>② 一次検証行動の選択</li> <li>③ 断定回避表現の選択</li> <li>④ 検証後対応手順の選択</li> <li>⑤ 最終確認必須業務の判別</li> </ol>	<ol style="list-style-type: none"> <li>① 個人情報該当性の判断</li> <li>② 入力不適切情報の判別</li> <li>③ 権利侵害リスク利用例の識別</li> <li>④ 適切な引用要件の選択</li> <li>⑤ 相談・記録・承認手順の選択</li> </ol>

# 教育プログラム——シラバス【科目S1】

## 演習の標準フロー

- ① [導入] 指導者が課題趣旨・前提・禁止事項・提出物・評価観点を明示する。
- ② [作業] 受講者がテンプレート（チェックリスト・記録様式）に基づいて作業し、操作ログ・判断根拠を作成する。
- ③ [提出] 受講者がLMSに成果物を提出し、必要に応じてGitに履歴・差分を残す。
- ④ [講評] GPT-TAの自動評価（事実性・一貫性・再現性）＋指導者レビューで総評・是正指示を行う。

コマ名	04 組織内AIポリシーと品質評価観点	05 プロンプトの基本練習と出力の見直し	06 データ保護と権利表示のミニ設計
コマ概要	組織内AIポリシーの最小構成を知り、守るべき事項を憶える。事実性・一貫性・再現性等の品質観点を把握し、簡易チェック表の活用法を学習する。作業記録の残し方、承認の流れ、見直しの頻度等を整理し、実務に繋げる形を理解する。	架空会議のメモを社内共有文に整形し、その要約文と作業項目を作成する。事実性・一貫性・再現性で点検し、根拠資料に照合して誤りと断定過多を是正する。再実行し、形式の揺れも確認する。不明箇所は「要確認」と明示し、差分と判断根拠と停止基準を記録して提出する。	架空案件を想定し、取扱いデータを「公開／社内／機密／個人情報」に区分して入力可否と条件を定める。「保存・共有」も明示する。引用と出典表示の書式、作業記録の必須項目、迷う場合の確認・承認手順を整理し、運用雛形として提出する。
科目内での当該コマの位置付け	倫理・権利の課題（03）を実務運用に落とし込む橋渡しのコマである。演習（05）で使うチェック表と記録様式の雛形を整備し、提出基準を充して評価対象になり得る最小限の品質確認の手順を共有する。	講義で学んだ基礎を自分で実践して確認するコマである。設計演習（06）に進む前に、目的定義→作成→確認→修正→提出の最小ループを体験し、記録と差分の残し方も習得する。	科目S1の総括として、安全利用と品質確認を小さな運用に落とし込むコマである。以降の科目やPBLで再利用できる最小の様式と手順を用意し、初学者も迷わずに作業できる状態の達成を目指す。
コマ形態	講義（オンデマンド視聴）	演習（ライブ配信）	演習（ライブ配信）
コマ内容	<ul style="list-style-type: none"> <li>○ 組織内AIポリシーの最小構成</li> <li>○ 品質観点の基本（事実性・一貫性・再現性）</li> <li>○ 作業記録と承認の流れ</li> <li>○ 見直しと改善の小さな循環</li> </ul>	<ul style="list-style-type: none"> <li>○ 目的の1文化と基本プロンプト作成</li> <li>○ 出力の読み取りと書き直し</li> <li>○ 出典確認と記録</li> <li>○ 直前直後の差分と提出物の整え方</li> </ul>	<ul style="list-style-type: none"> <li>○ データ分類と持出禁止の線引き</li> <li>○ 引用と帰属の書式</li> <li>○ 作業記録の必須事項</li> <li>○ 相談先と承認の流れ</li> </ul>
キーワード	AIポリシー 遵守 事実性 一貫性 再現性 チェックリスト 作業記録 承認フロー	目的定義 プロンプト 要約 箇条書き 検証優先 修正ループ 差分 停止基準	データ分類 公開／社内／機密 PII（個人識別可能情報） 疑似化 引用表示 帰属 記録事項 エスカレーション
到達目標	<ul style="list-style-type: none"> <li>◇ 組織内AIポリシーと品質観点（事実性・一貫性・再現性）の位置付けを理解し、その目的を説明できる。</li> <li>◇ 作業記録や承認フローが必要になる理由を、運用上のリスクに関連付けて説明できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ 目的と条件を明示したプロンプトを作成し、出力を品質観点で点検して是正を実施できる。</li> <li>◇ 改善前後の差分と判断根拠を作業記録として整理し、停止基準を明示できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ 取扱いデータを区分し、生成AIへの入力可否と条件を判断して整理できる。</li> <li>◇ 引用・帰属・相談先を含む最小限の運用ルールを作成し、記録項目を明示できる。</li> </ul>
理解度確認テスト	<ol style="list-style-type: none"> <li>① AIポリシー必須要素の選別</li> <li>② 事実性・一貫性・再現性の対応付け</li> <li>③ 作業記録の必須要素の選別</li> <li>④ 品質問題時の一次対応の選別</li> <li>⑤ 例外利用時の承認要件の選別</li> </ol>	<ol style="list-style-type: none"> <li>① 目的適合プロンプトの選別</li> <li>② 検証優先情報（数値・固有名詞等）の特定</li> <li>③ 修正ループ手順順序の選別</li> <li>④ 提出・共有に必要な証跡項目の選別</li> <li>⑤ 改善終了（停止）基準の選別</li> </ol>	<ol style="list-style-type: none"> <li>① データ分類（公開／社内／機密）の判別</li> <li>② PII取扱い安全措置の選別</li> <li>③ 引用・帰属表示の適切性判断</li> <li>④ 監査対応の記録項目の選別</li> <li>⑤ エスカレーション先と手順の選別</li> </ol>

# 教育プログラム——シラバス【科目S2】 前半

## ▼【科目S2】 プロンプト設計とLLM活用（講義6コマ・演習4コマ・実習2コマ）

LLMの内部機構とAPI実装を理解し、出力評価と再プロンプトによる改善ループを通じて高品質の出力を得るプロンプト設計とRAG（文書検索で根拠を附与して回答精度を高める手法）構築を習得する。

コマ名	01 LLMの仕組とプロンプトの基本形	02 基本プロンプトでの結果の揃え方	03 出力評価の基本——よい答えの見分け方
コマ概要	LLMは、入力文から次の語を予測して文章を生成する。トークン（文章を分割した単位）・入力長・推論設定が出力の揺れに影響する点を把握し、目的・条件・出力形式を明示した基本プロンプトを作成した上で、入力制約を守った入力・設定・出力の対応関係を理解する。	架空会議のメモを題材にし、目的別の要約等を生成するプロンプトを複数案作成する。複数回実行で揺れを観察し、条件追加と形式固定で狙い通りの安定出力に整える。入力・推論設定・出力と差分、改善理由を作業記録に残して所定形式で提出する。	事実性・一貫性・再現性の3観点で出力を点検する。数値や固有名詞等の検証優先箇所を見付け、誤りの型を分類して出典確認と断定回避を行う。品質ゲート最低水準と停止基準を定め、判断根拠を作業記録に残す。次の演習で改善に繋げる。
科目内での当該コマの位置付け	科目S2の導入として、設計パターン・改善ループ・API・RAGに共通する語彙と記録作法を共通化するコマである。演習（02）で出力比較と差分を扱う前提を整え、揺れを前提にして判断する基礎を早期に確立する。	01で学んだ基本プロンプトを実際に記述して試す最初の演習コマである。この後に学ぶ改善ループやAPI実装が必要になる、出力形式の固定、再現性の確認、差分記録の作法等を、提出まで含めて先に確立する。	02で作成した出力を良否判定できる状態にするコマである。演習（04）で再プロンプト改善を回すため、評価観点・検証優先・停止基準・記録事項を統一する。この後に学ぶRAGやAPIでも同じ軸で点検する。
コマ形態	講義（オンデマンド視聴）	演習（ライブ配信）	講義（オンデマンド視聴）
コマ内容	<ul style="list-style-type: none"> <li>○ LLMの生成の流れ（確率生成・トークン）</li> <li>○ 入力長と設定が出力に与える影響</li> <li>○ 基本プロンプト（目的・条件・出力形式）</li> <li>○ 入力制約と記録（入力・設定・出力）</li> </ul>	<ul style="list-style-type: none"> <li>○ テンプレートに基づくプロンプト作成</li> <li>○ 反復実行による揺れの観察</li> <li>○ 条件追加・形式固定による調整</li> <li>○ 差分記録の作成と提出</li> </ul>	<ul style="list-style-type: none"> <li>○ 3観点（事実性・一貫性・再現性）の点検</li> <li>○ 検証優先箇所（数値・固有名詞等）の特定</li> <li>○ 品質ゲート最低水準と停止基準の設定</li> <li>○ 判断根拠の作業記録（共有前提）</li> </ul>
キーワード	生成AI LLM トークン 入力長 推論設定 プロンプト 出力形式 記録	テンプレート 目的 条件 出力形式 反復実行 再現性 チェックリスト 差分	事実性 一貫性 再現性 検証優先 品質ゲート 停止基準 断定回避 作業記録
到達目標	<ul style="list-style-type: none"> <li>◇ LLMの内部構造と入力・設定が出力に与える影響を理解し、設計時に留意すべき点を説明できる。</li> <li>◇ 出力の揺れが生ずる理由を理解し、安定化が必要になる背景を説明できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ 同一入力に対する出力の揺れを観察し、条件追加や形式固定を適用して再現性を高められる。</li> <li>◇ 複数回実行の結果を比較し、その差分と改善理由を記録できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ 出力評価の観点（事実性・一貫性・再現性）を理解し、各々が必要になる理由を説明できる。</li> <li>◇ 品質ゲートと停止基準の考え方を理解し、演習での適用方法を説明できる。</li> </ul>
理解度確認テスト（出題観点）	<ol style="list-style-type: none"> <li>① トークン概念の識別</li> <li>② 入力長と出力揺れの関係の把握</li> <li>③ 基本プロンプト要素の判定</li> <li>④ 推論設定変更の影響の推定</li> <li>⑤ 記録事項と入力制約の選択</li> </ol>	<ol style="list-style-type: none"> <li>① テンプレート必須要素の判定</li> <li>② 出力形式指定の適否の判断</li> <li>③ 揺れ低減の修正方針の選択</li> <li>④ 差分記録事項の選択</li> <li>⑤ 入力制約遵守の判定</li> </ol>	<ol style="list-style-type: none"> <li>① 3観点の対応付け</li> <li>② 検証優先箇所の選別</li> <li>③ 品質ゲート最低水準の判定</li> <li>④ 停止基準設定方針の選択</li> <li>⑤ 判断根拠記録項目の選択</li> </ol>

# 教育プログラム——シラバス【科目S2】 前半

## 【科目S2】 後半 (07~12) 各コマ名 ※ 次年度作成

- |                             |                               |
|-----------------------------|-------------------------------|
| 07 [講義] APIでのLLMの呼出し        | 10 [演習] 簡易RAGによる引用付き回答の作成     |
| 08 [実習] 雛形コードでのAPI呼出し実行     | 11 [講義] コスト・レイテンシ・スロットリング     |
| 09 [講義] RAGの基礎——文書を根拠にして答える | 12 [実習] 統合ミニ課題——プロンプト・API・RAG |

コマ名	04 再プロンプトによる改善ループ	05 プロンプト設計パターン	06 パターン組合せによる業務プロンプトの作成
コマ概要	誤りを含む出力を業務利用に堪える水準に引き上げることを目的とする。評価表で問題点を分類し、一次情報で照合して断定回避も含めて再プロンプトで是正する。差分と根拠を記録し、品質ゲートと停止基準を充した時点で提出する。	役割付与、分割統治、手順の段階化、出力例提示等の基本パターンを学習する。曖昧な要求を分解して追加質問を引き出す方法も扱い、誤誘導を避ける制約条件の書き方を習得する。再利用テンプレートとして整える観点も確認し、次の演習で効果を比較する。	問合せ対応を題材にし、役割付与や分割統治等を組み合わせて業務プロンプトを設計する。二案を作成して品質と再現性を比較し、採択理由と注意点を短文で整理する。改善ループを一度回し、出力形式を固定して差分と根拠を記録し、提出する。
科目内での当該コマの位置付け	03の評価軸を用いて改善ループを手順として回す演習コマである。この後に学ぶ設計パターン活用やAPI実装でも同じ改善作法を使うため、差分と根拠の記録まで含めて型を確実に固定する。	改善ループ(04)で見えた課題に対し、修正の引出しを増やすコマである。次の演習(06)でパターンを組み合わせ、品質と再現性が上がる構造を比較して選ぶため、型と注意点を本コマで整理する。	05の設計パターンを実際の課題に適用して定着させる演習コマである。作成したプロンプトはこ実習(08)のAPI呼出しで流用するため、出力形式の固定と記録様式を本コマで確実に統一する。
コマ形態	演習(ライブ配信)	講義(オンデマンド視聴)	演習(ライブ配信)
コマ内容	<ul style="list-style-type: none"> <li>○ 問題点抽出と分類(誤り・不足・冗長等)</li> <li>○ 再プロンプトの修正方針と適用</li> <li>○ 一次情報照合と断定回避への修正</li> <li>○ 差分・根拠記録と提出判定</li> </ul>	<ul style="list-style-type: none"> <li>○ 役割付与と前提条件の与え方</li> <li>○ 出力例提示と出力テンプレート化</li> <li>○ 段階化(手順化)と分割統治の考え方</li> <li>○ 追加質問と制約条件(誤誘導回避)</li> </ul>	<ul style="list-style-type: none"> <li>○ パターン選択と組合せ設計</li> <li>○ 複数案の比較評価(品質・再現性)</li> <li>○ 改善ループ併用と最終案決定</li> <li>○ 設計意図整理と提出(記録添付)</li> </ul>
キーワード	改善ループ 再プロンプト 評価表 出典照合 断定回避 品質ゲート 停止基準 差分記録	役割付与 出力例 テンプレート 段階化 分割統治 追加質問 制約条件 誤誘導	パターン組合せ 業務課題 出力形式 再現性 改善ループ 比較評価 設計意図 記録
到達目標	<ul style="list-style-type: none"> <li>◇ 出力の問題点を把握・分類し、出典確認を行った上で再プロンプトによる是正を実施できる。</li> <li>◇ 品質ゲートと停止基準に基づいて改善を終了する判断を行い、その改善判断の根拠と差分を記録できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ プロンプト設計パターン(役割付与・分割統治・段階化等)を理解し、使い分けの考え方を説明できる。</li> <li>◇ 誤誘導を避けるための制約条件や追加質問の役割を説明できる。</li> </ul>	<ul style="list-style-type: none"> <li>◇ 設計パターンを組み合わせる業務想定に適合するプロンプトを作成し、出力形式を固定できる。</li> <li>◇ 複数案を比較評価し、その採択理由と設計意図を記録して整理できる。</li> </ul>
理解度確認テスト	<ol style="list-style-type: none"> <li>① 問題点分類の選択</li> <li>② 再プロンプト修正方針の選択</li> <li>③ 出典照合要否の判定</li> <li>④ 停止基準到達の判定</li> <li>⑤ 差分・根拠記録の要点の選択</li> </ol>	<ol style="list-style-type: none"> <li>① 役割付与の適用場面の選択</li> <li>② 出力例提示効果の判定</li> <li>③ 段階化・分割統治の使い分け</li> <li>④ 追加質問設計意図の判定</li> <li>⑤ 制約条件記述の適否の判断</li> </ol>	<ol style="list-style-type: none"> <li>① パターン組合せ順序の選択</li> <li>② 出力形式統一要点の判定</li> <li>③ 比較評価観点の選択</li> <li>④ 改善ループ適用の判断</li> <li>⑤ 提出記録の充足の判定</li> </ol>

## ▼ 講義映像 (S1: 01, 02, 03, 04, S2: 02, 04, 06)

学校法人電子学園 日本電子専門学校  
AI補完型ITエンジニア養成プログラム

【科目S1】生成AIリテラシーとAI倫理  
01 生成AIの基礎と安全利用の前提

### 要素1 & 2: 目的と条件

#### 目的(役割とタスクの定義)


「あなたはシニアJavaエンジニアです」

役割を与えることで、  
専門的な視点や用語選びを引き出す。

#### 条件(制約事項とコンテキスト)

「Java 17, Spring Bootを使用」  
「初心者向けに」

技術スタックやトーンを指定し、  
回答の精度を高める。



学校法人電子学園 日本電子専門学校  
AI補完型ITエンジニア養成プログラム

【科目S2】プロンプト設計とLLM活用  
01 LLMの仕組みとプロンプトの基本形

### ② LLMの生成の仕組み (トークンの予測と自己回帰)

LLMの生成は、驚くほど地道な「繰り返し作業」です。

▶ステップ1 (トークン化) ▶ステップ2 (確率分布の計算)



▶ステップ3 (トークンの選択) ▶ステップ4 (自己回帰)

入力プロンプト「吾輩は猫である」

→「吾輩は猫である。名前はまだない」

→次にくるトークンが「名前はまだない」の確率は90%、

→次にくるトークンが「名前はミケ」である確率は5%……



## ▼ 自習問題 (S1: 01-06, S2: 01-06 各コマ5問)

### Q02 【S1-01】 空欄補充

プロンプト設計の説明として「目的・条件・( ) (字数・箇条書き等)を明示する」がある。空欄に入る最も適切なものは次のうちどれか。

- ① 出力形式      ② 入力制約      ③ 対象読者      ④ 参照資料

### Q05 【S1-01】 正誤判定

「生成AIの出力変動を抑えるには、出力形式と条件を固定することが有効である」という文の内容は正しいか、誤っているか。

- ① 正しい      ② 誤っている

### Q32 【S2-01】 適肢選択

システム開発で用いる基本プロンプト要素として最も適切なものは次のうちどれか。

- ① 目的・条件・出力形式      ② 目的・装飾・語尾  
③ 条件・余白・色      ④ 形式・比喩・絵文字

### Q34 【S2-01】 正誤判定

「推論設定の変更は、LLMの出力の揺れや長さに影響し得る」という文の内容は正しいか、誤っているか。

- ① 正しい      ② 誤っている

# 教育プラットフォーム——仕様

## アプリケーション アーキテクチャ



### フロントエンド (React)

コンポーネントベース（独立したUI部品）を組合せてUIを構築しているため、拡張性が高い。  
JSX (JavaScript XML) + CSS

- ▶ 教員： ダッシュボード、受講者管理、スキル評価基準作成、問題作成、テストセット作成、教材管理、コース作成（eラーニング）、進捗管理、テスト結果管理、インサイト分析
- ▶ 受講者： テスト受験、コース受講、テスト結果確認、スキル分析、AIフィードバック

### バックエンド実行環境 (司令塔)

フロントエンドからのリクエストを受けてデータ処理の流れ（ビジネスロジック）とAI連携（問題生成・採点・改善提案）を実行。

- ▶ Firebaseのデータベース機能（Cloud Firestore）との連携  Cloud Firestore
- ▶ 生成AI（Gemini）とのAPI連携（AIによる問題生成の指示、インサイト分析の指示）  Gemini
- ▶ Cloud Run（次項）への指示（プログラミングコードの実行と採点）

### 採点サーバ (Cloud Run) Cloud Run

バックエンドからの指示を受けて隔離された仮想的な実行空間（サンドボックス）でプログラミング課題の安全な実行と採点を行う。

- ▶ Python実行環境（標準ライブラリ（ファイル操作・ネットワーク通信・計算等のタスク用）を含む）を利用
- ▶ pandas（統計）・numpy（多次元配列）をインストール
- ▶ scikit-learn（機械学習）・matplotlib（グラフ描画）をインストール

## プラットフォーム インフラストラクチャ

Google Cloud上でFirebase（認証・DB・ストレージ）等のマネージド基盤を中核として採用



### データベース (NoSQL) 機能

非構造化データ（テキスト・マルチメディアデータ）や大量のデータ等のリアルタイム高速処理が可能。

#### 全てのアプリケーションデータの格納

- ▶ 受講者データ（ユーザー情報と役割（教員/受講者））
- ▶ スキル評価基準（BASIS）
- ▶ 問題（AI作成/手動作成）、テストセット、採点結果
- ▶ コース、コース進捗データ

☞ 以上のデータをフロントエンドから呼び出す。

### ユーザー認証機能

Googleが提供するセキュアな環境で運用。

#### 認証データの格納

- ▶ 教員と受講者のデータ（氏名・メールアドレス・役割）

☞ メールアドレス等を識別子として認証し、氏名・役割をプロフィール情報として保持する。

### クラウドストレージ機能

あらゆるファイル形式に対応。

#### 教材（テキスト+マルチメディアファイル）の格納

- ▶ テキスト・PDF・Office系ファイル等
- ▶ 画像・音声・動画等

☞ 格納されたデータを使用してeラーニングのコース作成に利用する。

# 教育プラットフォーム——UI 例



ダッシュボード



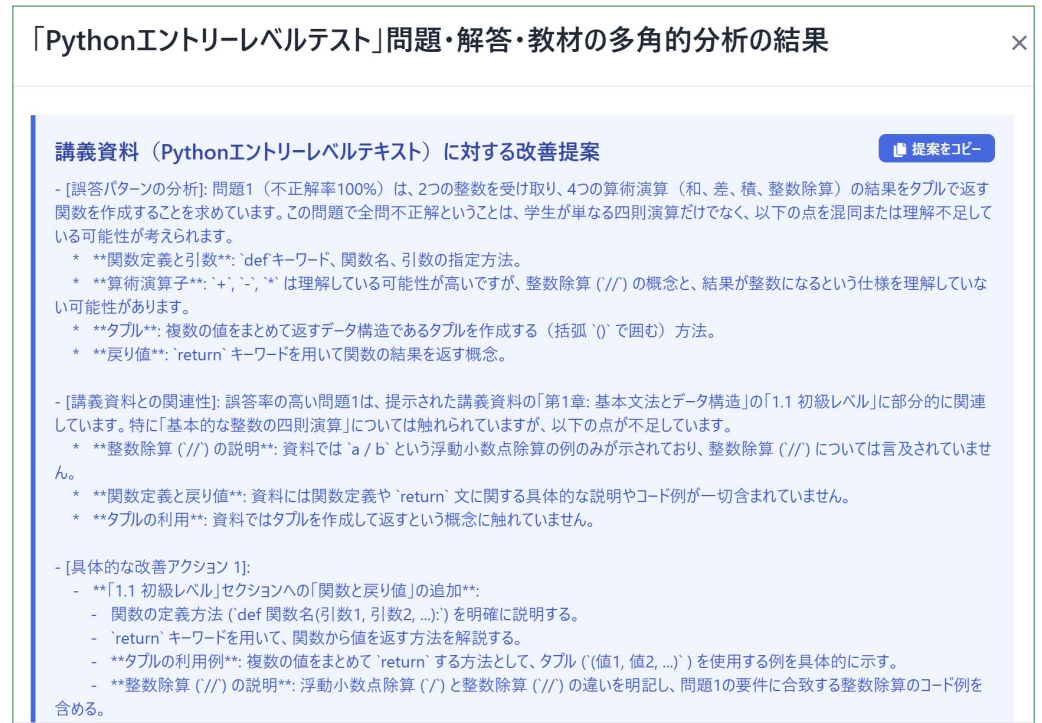
スキル評価基準の作成・編集



スキルマップの表示



AIによるプログラミング問題の生成



受講者の取組からの教育コンテンツの分析と改善提案



# 教育プラットフォーム——設計書 骨子

項目	概要
コーディング規約 (命名規則を含む)	ソースコードの記述ルール
機能一覧	システム上の全ての機能の詳細 ▶
処理機能記述	各機能の処理ロジック
画面一覧	全ての画面の名称と概要
画面仕様	各画面のデザインや操作方法等
画面遷移図	各画面への移動経路
項目ラベル名一覧	入力欄や表示項目の一覧
エラーメッセージ	表示されるエラーメッセージと対処法
ファイル一覧	入出力されるファイルの一覧
ファイル仕様	入出力されるファイルのデータ構造
外部インターフェース一覧	外部システムと連携する際の通信方法・API等の一覧
外部インターフェース仕様	外部システムとのデータの送受信形式・項目や、APIの呼び出し方法
バッチ処理一覧	定期的に自動実行されるバッチ処理の名称や実行タイミング
バッチ処理仕様書	各バッチ処理の具体的な実行内容・処理ロジックや、エラー時の動作等の定義
データモデル (コレクション/ドキュメント) 一覧	データベースに作成されるテーブルの一覧
データモデル定義	各テーブルの項目 (フィールド) 名、データ型の定義
セキュリティ仕様	セキュリティ上の脆弱性対策 (XSS・CSRF等) や、アクセス制御等のルール
アプリケーション仕様	ログの出力形式・文字コード・実行環境 (OS・ミドルウェア) 等、アプリケーション全体の共通的な技術要件

受講者用画面		教員用管理画面	
ダッシュボード	公開中のコース・テスト・結果・スキルを表示	ダッシュボード	システム全体の統計情報を可視化
コース一覧	公開中の全てのコースを一覧表示	受講者管理	受講者データのCRUD (作成・読取・更新・削除)
コース受講	選択したコースを受講する	教材バンク	コース (eラーニング) 用教材のCRUD
テスト一覧	受験可能な全てのテストを一覧表示	コース管理	コース (eラーニング) のCRUD
テスト結果	全てのテスト提出履歴を一覧表示	コースプロGRESS	受講者のコース進捗状況の一覧表示
スキルマップ	スキル習熟度を可視化	スキル管理	スキル評価基準のCRUD
		問題作成	AIと手動による問題作成 (プログラミング+通常) プログラミング問題を採点サーバで検証
		問題バンク	Firestoreに保存された問題の表示・CRUD
		テストセット管理	テストセットのCRUD+公開・非公開設定
		テスト結果	全受講者の全テスト提出結果を一覧表示 テスト結果詳細とスキルマップの表示
		分析	テスト結果・講義資料をAIで分析 → 改善提案 (インサイト) を生成

項目	機能概要
認証 (ログイン) ルール	Googleが提供するFirebase Authenticationを使用する。メールアドレスとパスワードによりログインを行う。
認可 (権限付与) ルール	firestore.rulesに記述したルールに基づき、サーバーサイドでのアクセス制御を実行する。 CRUD権限のうち、Create (登録)・Update (更新)・Delete (削除) は、原則として教員ロール (teacher) を持つユーザーだけに許可される (一部を除く)。 教員専用API (Cloud Functions (Callable)) でのonCall関数 (createUser (受講者登録)・generateTest (問題作成等)) は、処理の冒頭でrequest.auth.uidを使用し、呼び出し元が教員ロール (teacher) であることを firestoreのusersコレクションに問合せ検証する。
コードのサンドボックス実行	受講者が提出したPythonコードは、executorサーバー (Google Cloud Run) 内のサンドボックス環境で実行される。child_process.exec (コマンド) は、タイムアウト (10秒) 付き、uuidで生成された一時ディレクトリ内 (隔離環境) で実行され、実行後に即時削除される。
XSS対策	フロントエンドはReactを採用しており、JSXによるDOMレンダリング時の標準的なXSSエスケープが適用される。つまり、クロスサイトスクリプティング (XSS) によるセキュリティ攻撃に対し、Reactがデフォルトで提供する自動的な防御機能が有効である。
APIキー保護	GEMINI_KEY等の機密情報は、firebase-functions/paramsとGoogle Cloud Secret Managerを使用して管理され、フロントエンドには公開されない。