

以下の論文は、ChatGPT をカスタマイズした「技術士二次試験三上塾 AI 論文作成ツール」による「概要論文」なので、一般的な合格論文と異なります。

勉強の進め方としては、以下の資料を元に論文の方向性を掴みつつ、概要表や出典などを参考に以下の概要論文を修正すると短時間で効率的な勉強ができます。以下の出典の URL をクリックしてもサイトが表示されない場合は、出典名をネット検索してください。

設問は以下のサイトか日本技術士会のサイトをご覧ください。

https://www.engineer.or.jp/c_categories/index02022.html

=====

以下では、**選択科目Ⅲの「問題解決能力・課題遂行能力」**を中心に、設問→課題→解決策→将来的懸念→対策が一本の論理でつながるよう整理します。令和 8 年度を選択科目Ⅲは、社会・技術の状況から複合的な問題を把握し、多様な視点から技術課題を抽出して、問題解決手法と遂行方を論理的・合理的に説明する能力が問われています。また、マニュアルでも「観点・技術課題→最重要課題→解決方法→懸念事項→対応策」という骨子で整理する考え方が示されています。

1. 設問のテーマの社会目的

この設問の社会目的は、

生成 AI によるソフトウェア開発の生産性向上を享受しつつ、人間中心の品質保証を確立し、社会が安心して利用できる高品質・安全なソフトウェアを継続的に提供すること

です。

生成 AI は、要件定義、コード生成、レビュー、テスト、テストデータ作成など、SDLC 全体への適用が進んでいます。IPA もこれらを主要な活用場面として整理しています。[\(情報処理推進機構\)](#)

したがって、この問題は単に「生成 AI で良いコードを書く方法」を問うものではありません。

AI によって開発主体・開発速度・成果物生成方法が変化したとき、従来の品質保証をどのように再構築するか

が本質です。

2. 出題者の意図

出題者は、大きく次の能力を確認しようとしていると考えます。

(1) AI 駆動開発によって品質管理上「何が変わるのか」を理解しているか

生成 AI は確率的に成果物を生成します。このため、従来の「人が設計し、人が実装したものをレビュー・テストする」という前提だけでは品質保証が難しくなります。

(2) 単なる問題点ではなく、技術課題へ変換できるか

例えば、

「AI が誤ったコードを生成する」

は問題です。

これを、

「AI 生成成果物の検証可能性の向上」

という達成すべき技術課題に変換する必要があります。

マニュアルでも、観点から技術課題を抽出し、その課題に解決策を対応させる構造が重要です。

(3) ソフトウェア工学の専門知識で解決できるか

単に「人が確認する」「教育する」では弱く、

- SQuaRE
- CI/CD
- 静的解析
- SAST/DAST
- テスト自動化
- SBOM
- Provenance
- Human-in-the-loop
- AI ガバナンス

などを使って、工学的な品質保証プロセスを提示する必要があります。

3. 設問の重要な箇所・注意すべき箇所

最重要語は、**「従来の開発手法から AI 駆動開発手法に転換」**です。

つまり一般的なソフトウェア品質論だけを書くのでは不十分です。「AI 導入によって従来の品質管理の何が成立しにくくなるか」まで分析します。

次に重要なのが、**「品質確保・管理のあり方」**です。

生成 AI そのものの性能評価だけではありません。要件、設計、コード、テスト、セキュリティ、構成管理、レビュー、リリースまで含めた**開発プロセス全体の品質管理**として考えます。

さらに（1）の**「多面的な観点」**は、単なる「品質面」「技術面」のような抽象語ではなく、課題を導出できる観点にします。

今回は、

- 成果物品質の観点
- セキュリティ・サプライチェーンの観点
- 開発プロセス・ガバナンスの観点

とすると、三つが重複しにくくなります。

そして（3）は、（2）の解決策を全部実施してもなお発生する**次の段階の問題**でなければなりません。

4. 論文作成上のポイント

この問題では、次の因果関係を崩さないことが重要です。

生成 AI 利用拡大

- 従来と異なる品質リスク発生
- 3 観点から技術課題を抽出
- 最重要課題を決定
- 3 つの解決策
- それでも残る将来的懸念
- 対策

特に課題部分では、マニュアルが重視する「課題の抽出過程」を書きます。つまり、

〇〇の観点で考えれば、△△の推進が課題である。具体的には、現在□□という問題があり、その要因は××である。

という形です。根拠となる事実・データ・技術的背景を示しながら課題へつなぐことが重要です。

5. 不合格者と合格者の思考の違い

設問を見た箇所	不合格になりやすい思考	合格につながる思考
AI 駆動開発	「生成 AI のメリットを書こう」	「AI 導入で品質保証の前提がどう変わるか」
品質	「バグを減らすこと」	「機能適合性、信頼性、セキュリティ、保守性等をどう保証するか」
課題	「AI が間違える」	「AI 生成成果物の検証可能性向上」
多面的	「技術・人・コスト」	「成果物品質・セキュリティ・開発統制」
解決策	「人がレビューする」	「品質ゲート、静的解析、自動テスト、SAST 等を CI/CD に組み込む」
AI 利用	「便利だから全面利用」	「AI 生成物も未検証成果物として扱う」
(3)	「情報漏えいが心配」	「自動化高度化による人間の検証能力低下という二次リスク」
全体	AI の知識を並べる	品質保証体系の再設計として論理展開する

6. 概要表

項目	内容
設問の概要（現状・問題）	生成 AI が要件定義、コード生成、レビュー、テスト等へ拡大し、開発の高速化・自動化が進む。一方、生成物には誤り、脆弱性、出典不明コード等が混入する可能性があり、従来の人間中心のレビューだけでは大量・高速に生成される成果物の品質保証が困難になる。
技術課題①	タイトル：AI 生成成果物の検証可能性の向上。観点：成果物品質。AI 生成コードにはもっともらしい誤実装等が混入し得るため、要求からコード・テストまで一貫した検証可能性を高めることが課題。
技術課題②	タイトル：AI 駆動開発におけるセキュリティ確保。観点：セキュリティ・サプライチェーン。生成コードへの脆弱性、秘密情報、危険な依存ライブラリ等の混入を防ぎ、成果物の由来を追跡可能にすることが課題。

項目	内容
技術課題③	タイトル：AI 利用を組み込んだ品質ガバナンスの確立。観点：開発プロセス・ガバナンス。利用モデルやプロンプト、生成条件等により成果物が変動するため、責任分界、利用基準、記録、品質判定基準を標準化することが課題。
最重要技術課題	AI 生成成果物の検証可能性の向上
最重要とする理由	AI 生成物の品質を客観的に検証できなければ、生成 AI による開発速度向上に対して品質保証が追従できず、不具合・脆弱性が後工程や運用環境へ流出するため。品質保証の基盤となる課題である。
解決策①	品質要求の明確化とトレーサビリティ確保。ISO/IEC 25010 の品質特性を用いて品質要求を定義し、要求—設計—コード—テストのトレーサビリティを確保する。AI 生成物にはモデル、プロンプト、生成日時等のメタデータを紐付ける。
解決策②	CI/CD への自動品質ゲートの実装。単体・結合・回帰テスト、コードカバレッジ、静的解析、SAST/DAST 等をパイプラインに組み込み、AI 生成コードを自動検証する。
解決策③	Human-in-the-loop によるリスクベースレビュー。AI だけで合否判定せず、高リスク部分を専門技術者がレビューする。重要度に応じてレビュー強度を変え、AI と人間を相互補完させる。
懸念事項	品質保証まで AI へ依存することで、人間の設計・レビュー能力が低下し、生成 AI と評価 AI が同種の誤りを見逃す「自動化バイアス・共通モード故障」が顕在化する。モデル更新によって過去の品質基準を再現できない懸念もある。
対策	独立した多層品質保証の維持。AI 生成系と評価系を分離し、決定論的テスト、静的解析、第三者レビュー、レッドチーミング等を併用する。またモデル・プロンプト・依存関係・評価結果の Provenance を保存し、定期的な人手レビューと技術者教育を継続する。

参考出典

1. 独立行政法人情報処理推進機構（IPA）「AI を用いたソフトウェア開発」－要件定義、コード生成、レビュー、テスト等への AI 利用を整理。[（情報処理推進機構）IPA「AI を用いたソフトウェア開発」](#)
2. NIST, “Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1)”－生成 AI 固有のリスク管理を AI ライフサイクル全体で扱う。[（NIST）NIST AI 600-1](#)
3. NIST, “Secure Software Development Framework (SSDF)” / SP 800-218・SP 800-218A－セキュア開発と生成 AI・基盤モデル固有の開発プラクティス。[（NIST](#)

[コンピュータセキュリティリソースセンター](#))

[NIST Secure Software Development Framework](#)

4. ISO/IEC 25010:2023, “Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model” – ソフトウェア製品品質を規定・測定・評価するための品質モデル。
([ISO](#))
[ISO/IEC 25010:2023](#)
 5. OWASP “Top 10 for LLM Applications 2025” – Prompt Injection、Sensitive Information Disclosure、Supply Chain 等の生成 AI 固有リスクを整理。([OWASP Gen AI Security Project](#))
[OWASP Top 10 for LLM Applications 2025](#)
 6. 経済産業省「AI 事業者ガイドライン 第 1.2 版」 – AI の開発・提供・利用に関するガバナンス等を整理。([経済産業省](#))
[経済産業省「AI 事業者ガイドライン」](#)
 7. ISO/IEC 42001:2023, “Information technology – Artificial intelligence – Management system” – AI マネジメントシステムの構築、実施、維持、継続的改善を規定。([ISO](#))
[ISO/IEC 42001:2023](#)
 8. SLSA v1.2 “Provenance” – ソフトウェア成果物について、どこで、いつ、どのように生成されたかを追跡可能にする考え方。([SLSA](#))
[SLSA Provenance](#)
-

7. 1800 文字程度の論文

以下は、指定された章立てにそのまま合わせた答案例です。

1. 多面的な観点からの技術課題の抽出

1) AI 生成成果物の検証可能性の向上

成果物品質の観点で考えれば、AI 生成成果物の検証可能性の向上が課題である。具体的には、生成 AI は要件定義、設計、コード生成、レビュー、テスト等の開発工程に急速に適用されている。一方、生成 AI は確率的に出力するため、もっともらしい誤実装や要求との不整合を含むコードを生成する可能性がある。また、大量のコードを短時間で生成できるため、人手レビューのみでは品質確認がボトルネックとなる。このため、要求から設計、コード、テストまでのトレーサビリティを確保し、AI 生成成果物を客観的かつ継続的に検証できる仕組みが必要である。

2) AI 駆動開発におけるセキュリティ確保

セキュリティ・サプライチェーンの観点で考えれば、AI 駆動開発におけるセキュリティ確保が課題である。具体的には、AI 生成コードには脆弱な実装、秘密情報、不適切な OSS や依存ライブラリが混入する可能性がある。また、生成されたコードの由来が不明確な場合、脆弱性発見時の影響範囲特定も困難となる。このため、AI 生成コードを含むソフトウェア構成を可視化し、脆弱性とサプライチェーンを継続的に管理する必要がある。

3) AI 利用を組み込んだ品質ガバナンスの確立

開発プロセス・ガバナンスの観点で考えれば、AI 利用を組み込んだ品質ガバナンスの確立が課題である。具体的には、生成結果は使用する LLM、モデルのバージョン、プロンプト、コンテキスト等によって変化する。このため、従来の成果物管理だけでは生成過程を十分に説明・再現できない。そこで、AI の適用範囲、責任分界、品質判定基準及び生成履歴を組織的に管理する必要がある。

2. 抽出した技術課題のうち最も重要と考える技術課題と解決策

私は「AI 生成成果物の検証可能性の向上」が最も重要な課題と考える。理由は、生成 AI による開発速度の向上に品質保証が追従できなければ、不具合や脆弱性が後工程や本番環境へ流出し、AI 駆動開発そのものへの社会的信頼を失うからである。

1) 品質要求とトレーサビリティの確立

品質要求を明確化し、生成物を追跡可能にすることが重要である。具体的には、ISO/IEC 25010 の品質モデルを活用し、機能適合性、信頼性、セキュリティ、保守性等について品質要求と合格基準を定量化する。さらに要求、設計、コード、テストケースを紐付け、生成に使用したモデル、バージョン、プロンプト等のメタデータを記録する。これにより、不具合発生時の原因追跡と再検証を可能にする。

2) CI/CD による自動品質ゲートの構築

AI 生成コードを機械的に検証する品質ゲートの構築が重要である。具体的には、CI/CD パイプラインに単体・結合・回帰テスト、コードカバレッジ、静的解析、SAST、DAST 及び依存関係スキャンを組み込む。基準未達のコードは自動的にマージを禁止する。これにより、生成速度が向上しても品質確認を継続的かつ再現可能に実施する。

3) Human-in-the-loop によるリスクベースレビュー

AI と人間を相互補完させるレビューが重要である。具体的には、決済、個人情報、認証等の高リスク機能については専門技術者によるレビューを必須とする。一方、低リスク部分は自動テスト・静的解析を中心とし、リスクに応じてレビュー強度を変える。これにより、AI の生産性と人間による判断の双方を活用する。

3. 解決策を実行しても新たに上がってくる将来的な懸念事項と対策

1) 懸念事項

懸念事項としては、品質保証の AI 依存による技術者の検証能力低下がある。具体的には、コード生成だけでなくテスト生成やレビューまで AI 化すると、人間が AI の出力を過信する自動化バイアスが生じる。また、生成 AI と評価 AI が類似モデルや学習データに依存した場合、同一の誤りを双方が見逃す共通モード故障が発生し、不具合が品質ゲートを通過する可能性がある。

2) 対策

対策としては、独立した多層品質保証の維持がある。具体的には、生成系 AI と評価系 AI を可能な限り分離し、決定論的な自動テスト、静的解析、専門技術者によるレビュー及びレッドチーミングを組み合わせる。また、モデル、プロンプト、依存ライブラリ、テスト結果等の Provenance を保存し、モデル更新時には回帰評価を実施する。さらに、技術者自身が設計・レビューを行う演習を継続し、AI を使いこなしながらも最終的な品質責任を人間が担える体制を維持する。

この答案で特に意識しているのは、「AI が危険だから人が確認する」という一般論ではなく、ソフトウェア工学として品質保証体系そのものを再設計するという方向です。選択科目Ⅲでは専門的学識だけでなく、問題解決・評価・コミュニケーションが評価対象であり、今回のように「現状→原因→課題→技術的解決→新たなリスク」まで一貫させることが重要です。

上記の様に、自分の受験部門・受験科目のわかりやすい過去問解説や概要論文を作って効率的に勉強したいという方は、以下の「技術士二次試験三上塾 AI 論文作成プログラム」を参考にしてください。

<https://suninfo.co.jp/g2gpt/>

技術士二次試験 三上塾