

第 11 回中級ソフトウェア品質技術者資格試験記述式問題の解説

※出題した問題の一部を解説付きで公開いたします。正しい知識の習得と理解の深耕にご活用ください。

※中級ソフトウェア品質技術者資格試験の記述式問題の採点においては、唯一の正解との適合のみをみるのではなく、受験者の意図を読み取って採点しています。

■穴埋め問題

空欄（ ）に入る適切な語句を解答用紙の該当箇所に解答せよ。

【解答用紙】 実際の解答用紙のスタイルです。

	①	②		①	②
問題26			問題31		
問題27			問題32		
問題28			問題33		
問題29			問題34		
問題30			問題35		

【問題】 保守の技法

(①) は、ソースコード中のまったく同じあるいは類似したコードを意味する。(①) があると、ソースコードの (②) 時にすべての (①) を (②) しなければならず、ソフトウェアの保守を困難にしてしまうため、注意が必要である。

【解答例】

- ①コードクローン
- ②修正、更新、改造、変更 等

【解説】

ソースコード中のまったく同じあるいは類似したコードをコードクローンという。ソースコード中に同じコードが散在すると、そのコードが誤っていた場合に、その同じコードをすべて探して修正する必要がある。しかも、類似したコードを異なる意味で使っている場合も考えられるため、単純な修正は危険であり、一つ一つ検討して修正しなければならない。このように、コードクローンは様々な問題を引き起こす原因となる。

コードクローンは大抵、開発者がソースコードをコピー&ペーストすることで発生する。開発時には、プログラムを構造化するよりも、単純にコピー&ペーストするほうが生産性が良いように感じられるかもしれないが、実際は逆である。今やソフトウェアを修正せずに使い続けることはほとんどなく、リリース後に必ず保守や機能追加が発生する。コードクローンがあると、そのような保守や機能追加のたびに、修正自体が難しく、また思わぬ箇所での障害発生などデグレートの危険性があるため、広範囲のレビューやテストが必要となる。コードクローンは、ソフトウェアライフサイクル全体へ悪影響をもたらすのである。

誤った解答の例としては、①冗長ソースコード、コピーコード、②ホワイトボックスなどがあつた。コードクローンは重要な考え方であり、用語とその意味をぜひ理解していただきたい。

【問題】 レビュー技法

FMEA 及び FMECA は、ディペンダビリティ予測において基本的な解析の手法として 使用される。一般に表を用い、影響や（①）を定量的に解析していく。その手順は、以下のとおりである。

- (1) 対象となるシステムの機能や部品などの中でシステムの信頼性に大きな影響を与えるものを抽出する。
- (2) これら機能や部品について（②）を洗い出し、その原因を調べる。
- (3) （②）ごとにシステムへの影響、発生確率を評価する。
- (4) 影響が大きなもの、発生確率が高いものから順に対策を検討する。

【解答例】

- ① 致命度
- ② 故障モード

【解説】

FMEA (Failure Mode and Effects Analysis : 故障モード・影響解析、またはフォールトモード・影響解析) 及び FMECA (Failure Mode, Effects and Criticality Analysis : 故障モード・影響及び致命度解析、またはフォールトモード・影響及び致命度解析) は、システム中のあるアイテムの故障モードに着目し、その原因調査、影響調査をすることで、システムの信頼性を解析する方法である。FMEA は、ソフトウェアの障害に着目して信頼性解析をする。FMECA は、FMEA に付加して、故障発生の確率および故障による影響の重大さの格付けを考慮する技法である。また、ディペンダビリティとは、「アイテムが、要求されたときに、その要求どおりに遂行するための能力」である。問題文中のディペンダビリティ予測とは、時間の経過に伴うシステムの使用状況の変化や、利用に伴うシステムの経年劣化などによる、信頼性への影響予測を意味している。

FMEA や FMECA の適用方法としては、例えば、ソフトウェアの開発や運用において過去に検出、発見された障害を抽象化し、致命度に応じて順位付けして、欠陥パターンとしてリスト化しておく。欠陥パターンには、欠陥ごとの原因や前提条件、起こりやすさ、検出可能性を含む。また、致命度は、利用者が被る影響や修正コストなどを考慮する。そして、ソフトウェア開発時において、リスト化した欠陥パターンがどのような場面で起こりうるかを想定し、予防、検出に役立てるのである。

誤った解答の例としては、①危害、原因、②バグ、ハザード、事象などがあつた。本問題は、今回の穴埋め問題で最も正答率が低かつた。

一方、国際規格に過度に依存することで生じる弊害にも目を向けておく必要がある。個々のソフトウェアやシステムそのもの、そして、それらに関係する開発組織は多様であり、常に進化していくものである。したがって、当事者の製品や組織運営の目的に応じて、国際規格に定められている包括的なライフサイクルプロセスを、当事者向けにテーラリングすることが必要である。規格に縛られてしまつては本末転倒である。ライフサイクルプロセスを標準化することによって、プロジェクトの混乱を防ぎ、一定の品質を確保する効果が期待できるが、定めた標準に安住することなく、変化する外部環境に合わない部分は、常に改善していくことが肝要である。

サービス開始まで 2 週間」という状況において、サービス開始判定を行う場合に、品質面で確認しておくべき事項を問う問題である。その際、これまでの開発における経緯や現システムの存在など、このプロジェクトにとっての不安な点をおさえた上で、サービスを開始したあとのリスクを想定する必要がある。また、システムが稼働し、サービスを開始した後には生じる問題とその対策についての理解も問うている。

解答のポイント：

情報システムのサービス開始直前に確認すべき項目は、以下のようにサービス開始日前後で時系列的に整理し、リストアップすると分かり易い。

《サービス開始前》

- ① 未完了の作業・部分に対する見通しや扱い
- ② 開発中に問題の多かった部分に対する最終的な念押し確認

《サービス開始当日》

現システムからの移行やシステムの切替作業など、稼働日当日に向けた準備状況の確認

《サービス開始後》

サービス開始直後、また開始後一ヵ月以内など短期間のうちに起こる可能性の高い障害への備え、準備状況の確認

懸念事項として挙げるべき、具体的な確認ポイントについて述べる。

《サービス開始前》の①に関しては、問題文に挙がっているシステム総合試験の完了見通しや残存バグの状況、未解決バグの扱いの他に、システム総合試験のバグの収束状況や試験項目の十分性の最終確認などが考えられる。②に関しては、プロジェクト混乱時期があったため、見切りで進めたと想定される工程の残課題の対応確認や、保守・運用グループからの問題提起による設計変更の反映確認、などが挙げられる。

《サービス開始当日》については、開始当日の作業の準備状況を確認する。特に移行については、現行システムが直前まで稼働していることが想定されるため、当日蓋を開けてみなければ分からない部分もあり、リハーサルによる十分な事前確認と、入念な準備が必要となる。

《サービス開始後》については、まずサービス開始直後に各種トラブルが発生する可能性が高いことから、その際の連絡体制・即時解析の体制が整っていることを確認すべきである。また、サービス開始直後に発生しやすいトラブルとして、性能、使用性の問題による問合せの多発などが考えられるため、それらに向けた対処として、性能テストの最終的な実施、および結果確認や、ユーザトレーニングの状況、もしくはサービスデスクの準備状況について確認しておく必要がある。

上記のような、サービス開始日前後に開発・運用チームが行う事項という観点の他に、開始判定そのものの基準の決定プロセスや整備状況、また判定の結果、開始が NG となった場合の顧客への説明プランが検討されているかなど、開始判定のマネジメントに関する回答についても、正答とした。

不十分な解答の特徴の例：

- ・ 問題文に挙がっている 3 例と、ほぼ同義の回答。
- ・ 本問題は「情報システムのサービス開始前」に関する出題であるが、「ソフトウェア製品の出荷前」を想定とした懸念となっている回答。
- ・ サービス開始前ではなく、計画段階、設計段階で確認する内容の回答。
例) 現行システムとの差異が洗い出され、テストされているか？
発生した不具合の原因分析が行われ、類似調査や影響範囲の確認がなされているか？
- ・ サービス開始判定における懸念事項として具体的ではない回答。
例) サービス開始後に発生しうるリスクについて対応策が検討できているか？

以上