代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
機能性	合目的性 ユーザ要求とソフトウェア仕様(又はシステム仕様)との合致度合いを高めるために講じる事例	妥当性(利用者側の要件)の確認(レビュー、テスト)に対するユーザとベンダとの役割分担		①ユーザ業務部門からの業務要件について、ユーザIT部門が内容を確認し、要件定義に着手するにあたりその十分度を判定。この後、ユーザIT部門が文書化された業務要件をベンダに提示、ベンダは不明点・疑問点を一覧化して、ユーザに確認。やりとりした内容について3者合意が取れた段階で要件定義に着手。この時点で未決事項については一覧化してユーザ・ベンダで共有し、以降のインプットとする。 **●工夫・対応の効果** ・要件が明確になり、後工程での手戻りの発生を防止する。 ・未決事項に対して課題管理として開発関係者で共有することができ、解決に向けた動きが期待できる。		○	○		○	○
				②要件定義書(システム化機能)は有識者(必要に応じベンダ)が作成し、ユーザIT部門と全ての内容についてレビューを行う。レビュー後、ユーザ業務部門が検証する。なお、非機能要件についても有識者(必要に応じベンダ)が資料を作成し、ユーザのインフラ構築部門および運用部門も確認する。 **●工夫・対応の効果** 非機能要件を早期に確定することでハードウェアの調達、ソフトウェアの選定が速やかに行え、後工程での手戻りの発生を防止する。		○	○		○	○
			28－④	③詳細設計書は要件定義書をもとにベンダが作成し、ベンダ内部レビューを経た後もしくはベンダ内部レビューに相乗りする形で、ユーザIT部門とのレビューを行い、要件が外部仕様に漏れなく展開されていることを確認する。さらにユーザ業務部門が検証することにより、要件のトレーサビリティを保証する。 **●工夫・対応の効果** 要件のトレーサビリティが確保でき機能の誤解、漏れの発生を早期に防止する。		○	○		○	○
				④テスト最終段階(総合試験)の試験項目、試験ストーリーについては有識者(必要に応じベンダ)が抽出・作成し、ユーザIT部門とレビューする他、必要に応じてユーザ業務部門の確認を受ける。試験結果についても、ベンダ検証の後、ユーザIT部門、ユーザ業務部門が検証する(検証範囲・レベルはユーザサイドと判断) **●工夫・対応の効果** 要件のトレーサビリティが確保でき機能の誤解、漏れの発生を防止する。		○	○		○	○
				⑤新規業務や運用が大幅に変更になるような場合は、ユーザ業務部門が企画・参加する試験を実施。ユーザIT部門およびベンダとの認識の齟齬を洗い出すのに有効。 **●工夫・対応の効果** 要件のトレーサビリティが確保でき機能の誤解、漏れの発生を防止する。		○	○		○	○
			3－①	⑥初期成果物に対するユーザレビューを実施し、成果物の記述水準・記述密度をユーザと合意することにより、以降作成する成果物に対する記述水準レベルでの指摘件数を削減する。 **●工夫・対応の効果** レビュー工数の削減、レビューによる手戻り工数の削減により、コストダウンが図れる。		○	○		○	
			1－③	⑦システムテスト仕様書の作成、システムテストはユーザ中心で行うことにより、業務面での確認を実施する(ベンダは、テストデータの作成等環境整備が中心となる)。 **●工夫・対応の効果** ・開発者が想定できない実業務に沿ったテストが可能となるため、残存欠陥を検出することができる。 ・利用者の立場に立ったテストを実施することで、運用開始後の仕様変更、追加工数が削減できる。		○	○		○	
				⑧開発ライフサイクルの選択、各ステップにおける代表タスクおよびスケジュール、工数、要員計画、レビューや承認のマイルストンを立案し、ユーザ・ベンダの双方でレビューしながら役割分担を確認する。 **●工夫・対応の効果** プロジェクト利害関係者が参加し開発プロセスを作成することで、開発プロセス全体を早い時期に見極めることができるため、開発担当者の役割分担が明確となり作業の重複を防止する。		○				
				⑨利用者向けの運用基準書作成においては、ユーザ企業の開発部門担当者だけでなく、運用部門担当者が同席した形態でのレビュー会を実施する。 **●工夫・対応の効果** 早期に運用、利用形態を見極めることで後工程での手戻り工数の発生を防止する。		○				
			3－① 2－①	⑩ユーザ要件定義工程において、業務主管部門とIT部門のそれぞれの部署が責任をもって作成しなければならないドキュメント類を定め、要件定義における検討ポイントや記載要領等を解説したガイドを準備し運用する。 **●工夫・対応の効果** プロジェクト参加者の役割分担、責任範囲が明確となるため、作業の重複を防止するとともにスケジュールの遵守が期待できる。		○				
				⑪アプリケーションオーナー制度(※1)を導入し、開発の各工程でのオーナーの役割を明確にする。オーナー制度を定着させるため、定期的な進捗やレビュー会議の都度、オーナー制度の説明を行い、オーナー向けのシステム開発研修(※2)を実施する。 **●工夫・対応の効果** システム開発へのオーナーの理解度・参画割合が高まったことで、トラブルが削減できる。 ※1 アプリケーションオーナー制度:システム開発はIT部門(サプライヤ)とビジネスサイドのアプリケーションオーナーとIT部門の共同作業であり、オーナーには要件の提示や各チェックポイントでシステム開発内容にコミットメントする役割があることを定めたもの。 ※2 IT開発の進め方についてのユーザ向け研修。システム開発のポイントをユーザ(オーナー)に理解してもらい、プロジェクトを円滑にすすめることを目的とする。		○	○			
				⑫実際の利用者が外部設計、結合テストのレビューアになる。 **●工夫・対応の効果** ・早期に利用形態を見極めることで後工程での手戻り工数の発生を防止する。 ・利用者の立場に立ったテストを実施することで、運用開始後の仕様変更、追加工数が削減できる。		○	○		○	
				⑬発注者ビューガイドラインを使用した成果物の明確化、ならびに、レビュー観点の統一を図ることにより、工数を削減する。 **●工夫・対応の効果** プロジェクト参加者の役割分担、責任範囲が明確となるため、作業の重複や漏れを防止する。また、レビュー観点が予め提示されることにより成果物の精度が上がり、スケジュールの遵守が期待できる。		○	○			
			33－①	⑭生産物レビューについて、ベンダとユーザ(IT部門、業務部門、運用部門)のレビュー担当範囲を定義する。また、各担当がレビュー結果に対する責任を宣言する意味で、レビュー報告書への検収捺印を行う。 **●工夫・対応の効果** レビュー担当者の役割分担、責任範囲が明確となるため、レビュー漏れの防止、レビュー品質の向上が期待できる。		○	○		○	

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
機能性	合目的性 ユーザ要求とソフトウェア仕様(又はシステム仕様)との合致度合いを高めるために講じる事例	暗黙要求(要件記述なし)の確認手段に関する取り決め	1－①	①要件に記載が漏れやすい下記の内容について、設計工程の早い段階で明文化する。 ・業界常識、顧客常識、顧客標準ビジネスルール。 ・利用者のスキル、操作錬度。 ・各種ログのサイズ、保存期間、データのバックアップに要する時間等の非機能要求。 **●工夫・対応の効果** 明文化することでプロジェクト参加者の機能の理解を早めることができるとともに、以降工程においても業務を意識した作業を行うため、設計の漏れ等が減少する。		○				○
				②現場利用者を参画させ、プロトタイプや画面モックアップにより要求機能の先だしを実施し、現物をもとに利用者が参画して仕様の確認を実施することを開発プロセスとして規定する。 運用管理等の非機能要件(バックアップ、セキュリティ、資源配布管理方式等)については共通的なフレームワークや実装基盤を標準化し、それらに関する技術実装に特化したレビューを実施することを規定しており、不備等の確認や合意形成を行う。 **●工夫・対応の効果** 上流設計段階でのフィージビリティ確保、手戻りの抑止が期待できる。		○	○			○
				③既存システムへのサブシステム追加や機能拡充の場合、既存システムのシステム上の制約事項やパフォーマンス要件を一覧化したチェックリストを使用し、フィージビリティチェックを実施したうえ、当該システムの有識者によるレビューを実施し、既存システムの枠組みから逸脱を排除する。 **●工夫・対応の効果** 既存システムの特性をプロジェクト参加者すべてが認識することにより、設計の漏れを防ぎ、テスト項目の設定工数の削減を図ることができる。		○				
				④非機能要件チェックシートを運用部門に必須提出とし、開発時に忘れがちな非機能に関する要件の明確化を図る。 **●工夫・対応の効果** 運用部門からの意見を取り入れることにより、非機能要件の漏れを早期に把握することができる。		○	○			
			1－①	⑤要件網羅、要件要素間矛盾および妥当性の観点から暗黙知による要求欠如、要求項目同士の矛盾および背景・スコープの不明確さなどを第三者要件定義診断を実施する体制を組織化する。 **●工夫・対応の効果** 診断実施により、要件要素の漏れや要素間の矛盾を早期に発見することができる。		○				
		仕様変更に対するユーザとベンダーとの合意		①仕様変更に関しては、「情報システムの信頼性向上のための取引慣行・契約に関する研究会」報告書のモデル契約書第4章「契約内容等の変更」を参考に、基本取引契約あるいは個別取引契約書、見積仕様書などで合意することを推進している。 実務面としては、分かりやすい詳細な仕様連絡にすること、仕様連絡書の運用ルールをプロジェクトキックオフ時に合意することを推進している。 **●工夫・対応の効果** ・契約書にて仕様変更の扱いについて合意しているため、スムーズな運用が図れる。 ・仕様連絡書にて仕様のやり取りをしているため、後になっての言った言わないの話がなくなり、契約折衝での揉め事がなくなる。	○					
			4－①	②大規模プロジェクトでは、要件変更は直接の関連部門だけでなく全体をとりまとめ、事務局の職位者の承認を必要とし、要件変更によるコストや工数、スケジュールへの影響を全体として管理する仕組みとする。 **●工夫・対応の効果** ・仕様変更の要否について検討する場として事務局が存在したため、開発者の工数負担を軽減させる。 ・費用対効果を明確にすることで、予算内での開発を推進することができる。 ・仕様変更については、事前に文書による合意を行ったうえで対応したため、後の揉め事を回避する。			○			
			4－①	③詳細設計書の納入(仕様の確定)以降発生する変更は全て仕様変更として位置づけ、追加発生する稼働およびその費用負担配分、取り込み時期等を協議し、双方合意後、規定された依頼文書の授受により発生から完了まで管理する(定例のユーザ・ベンダ間の進捗会議において状況報告を行う)。 **●工夫・対応の効果** ・事前に文書による合意を行ったうえで対応するため、後になって揉めることがなくなる。 ・仕様変更の進捗についても逐一報告しているため、本体への影響についても合意を得ることができる。			○			
		ユーザ要件に対する設計書の必要十分性(トレーサビリティ)の検証	1－②	①要件定義書から要件を一覧化し、要件ごとにIDを付与し、以後、設計書ならびに試験仕様書においてこのIDをベースに詳細化(IDの枝番付与等)しながらトレーサビリティを確保する。 加えて、各種チェックリストの活用も推進する。 **●工夫・対応の効果** 詳細設計局面での要件との整合性が可視化されるため、漏れ等の把握が容易となる。		○	○		○	
			2－②	②要件定義書・基本設計書にユーザ要件がすべて盛り込まれているか、誤解している内容がないかを内部レビュー観点とし、内部レビューを実施した後、最終的にユーザ・ベンダ両者参加の設計書レビューにて確認する。 **●工夫・対応の効果** 内部レビュー観点に加えたことにより、設計書作成者も要件の漏れを意識することになり、結果として品質向上に役立つ。		○	○			
				③ユーザ要件を「機能要求」と「非機能要求」に分け、「機能要求」については「要件定義書」と「基本設計書」の記述を相互に記述項目の番号で対応付けることにより、必要十分性を検証する。 **●工夫・対応の効果** 機能要件については、詳細設計局面での要件との整合性が可視化されるため、漏れ等の把握が容易になる。		○	○			
				④上流工程における業務要件確認プロセスにおける取扱ルールの策定 業務要件、システム要件が具体化されているかまた、検討の積み残しの有無、懸案事項管理がされているか、各部門・関係箇所間の合意が形成されているか等をチェックリストをもとに、次工程の受託の可否判断のレビューを実施する。 **●工夫・対応の効果** 次工程への課題やその解決すべき時期等が明確になり、後工程で発生する手戻りやリスクが抑制できる。		○	○			
	正確性 ソフトウェア仕様(又はシステム仕様)とソフトウェア実装(又はシステム実装)との合致度合いを高めるために講じる事例	計算精度やデータ精度要求の実装確認	5－①	①丸めの誤差、情報落ちなどで微妙な誤差を防止するため、そもそもの法制度の趣旨や解釈を正しく理解することと合わせて、テストでの確認を実施する。また過去事例を反映したチェックリストに基づいて作業チェックを実施することにより、同様の誤りの発生を未然に防止する。 ・XXヶ月前の集計結果などで起点日の考え方(方端計算、両端計算の違い等)を確認する。 ・税制変更(例えば消費税)などで、端数(少数以下)の処理を確認する。 **●工夫・対応の効果** 端数誤差による障害発生が抑制できる。					○	○
				②ドキュメントが整備されていないプログラムについて、ソース解析機能により、同じ内容を保持している変数を複数プログラムに渡ってトレースしたり、DBを介した情報のつながりを追いかける機能を利用することで変数間の関係などを可視化し、漏れのない作業を行う。ソース解析により、複雑なデータ連携を持つプログラム連携部分を抜き出し、重点的に退行テストを行い、効率化を図る。 **●工夫・対応の効果** 対応漏れによる障害発生が抑制できる。				○		○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
機能性	正確性 ソフトウェア仕様(又はシステム仕様)とソフトウェア実装(又はシステム実装)との合致度合いを高めるために講じる事例	計算精度やデータ精度要求の実装確認	5－①	③設計書に計算するパラメータ(※1)の定義、計算順序、計算途中で発生する端数の取り扱いや桁落としを明記してユーザと合意する(開発言語のバージョンアップにより、同じパラメータ(※2)でもバージョンアップ前後で結果が異なる非互換が発生するが、システムで扱う精度内では問題とならないことを確認した上で対処する)。また過去事例を反映したチェックリストに基づいて作業チェックを実施することにより、同様の誤りの発生を未然に防止する。 ※1　利子計算における「元本」、「利率」、「利子計算期間」などの計算要素のこと ※2　コンパイル時に指定するコンパイラの各種オプションのこと **●工夫・対応の効果** 端数誤差による障害発生が抑制できる。		○	○			○
				④料金計算の改定要件に対し、四捨五入のタイミングの違いにより、誤差が発生する。合計してから率をかけるか、個別に率をかけて合算するかで誤差が生じるため、対応方法を事前に確定する。また過去事例を反映したチェックリストに基づいて作業チェックを実施することにより、同様の誤りの発生を未然に防止する。 **●工夫・対応の効果** 対応誤りによる障害発生が抑制できる。				○		○
		システム利用マニュアル記述の正確性向上施策		①マニュアルは、システム運用のためのものと、システム利用のためのものがある。システム運用のためのものは、開発部門が作成する。システム利用のためのものは、利用部門が作成する。機能追加・改修、システム構成変更が発生した場合は必ず更新する。 **●工夫・対応の効果** マニュアルの利用者が作成することで、マニュアルの精度が上がり、保守運用工数および利用者からの問い合わせ工数の削減に寄与する。			○			○
			7－③	②マニュアル作成後、実環境を使用した試験によりマニュアルの精度を検証する。 **●工夫・対応の効果** マニュアルの精度が上がり、問い合わせ工数の削減、および、誤操作の防止に寄与する。			○			○
			6－① 7－③	③改良開発後のユーザ誤操作や誤入力が散見していたため、改良開発にて「マニュアル改訂作業」を必須とする。 **●工夫・対応の効果** ユーザの誤入力や誤操作の発生頻度は減少する。			○			○
				④汎用ソフトウェアのマニュアルにはパラメータの設定値のみでなく、設定根拠や算出式を記述する。 **●工夫・対応の効果** ・システムパラメータの設定値の根拠が不明確だと、システムを修正する際に再設計をしなければならず、誤りを作りこみやすいが、これを未然に防止することにより、製品品質を高める。 ・パラメータの設定工数の削減に繋がるとともに、改修の影響範囲が明確になるため、調査工数の削減が見込める。			○			○
				⑤汎用ソフトウェアの開発・改修においては、標準化の専門グループが構成管理システムを顧客別に構築し、開発・マニュアル作成を行う。 **●工夫・対応の効果** 構成管理を徹底することにより、バージョンの取り違えの問題等減少し、結果としてコスト削減に繋がる。			○			○
		検証(工程間整合)確認に対するユーザとベンダとの合意	8－②	①ユーザにフェーズ毎のプロセス品質(計画・実績)、残存欠陥等含めたカットオーバ基準を公開し事前に合意する。 **●工夫・対応の効果** ・次工程に着手する前に品質の計画値と実績値の差異分析を行い、ユーザと合意形成を行うことができる。差異が大きい場合、レビュー精度の見直し等早期対策を講じられるため、結果として高品質の製品の提供に繋がる。 ・残存欠陥についてユーザと合意を得ることにより、品質目標とテストコストとの整合を図る。	○					
				②各工程の計画書(※1)およびベンダ側で実施した品質評価結果(※2)をユーザに報告し、ユーザからの質疑に回答する。ベンダの品質指標(試験密度、欠陥密度等)は計画書および報告書に記載し、公開する。 **●工夫・対応の効果** 次工程に着手する前に品質の計画値と実績値の差異分析を行い、ユーザと合意形成を行うことにより、品質評価指標の予実差異に起因する懸案に対し早期対策を講じることができ、結果として高品質の製品の提供に繋がる。 ※1　当該工程の目的・目標、作業方針、スケジュール(主要作業の期間、主要マイルストンを明示)、作業体制(主要作業の責任者を明示)、作業内容の概要(作業の指針となるドキュメント、特に留意すべき事項、使用する環境・ツールなど)、品質指標値、管理体制、会議体などを記載。 ※2　評価対象工程における ・実績値(試験密度、欠陥密度など)と品質指標値と照らした定量評価 ・定量評価を補完する定性評価(試験内容や見た体制、要員スキルの面の評価など) ・摘出した欠陥の分析結果とそこから展開して実施した品質向上対策の実施内容と結果およびその評価 ・開発上のペンディング事項の解消状況と残存しているペンディング事項の評価 ・仕様変更の発生・取り込み状況など					○	
			8－②	③工程完了報告会を実施し、開発規模、試験密度、レビュー指摘件数、障害件数をユーザに報告。ユーザより承認を得ることで次工程の作業に着手する。当該工程にて保留となった事項については申し送り事項として、別途管理し解決の都度ユーザに報告する。 **●工夫・対応の効果** ・次工程に着手する前に開発規模の増減等についてユーザと事前合意できるため、開発費用の増減もしくは機能の削減を調整することができる。 ・必要以上に指摘件数・障害件数が多い場合、次工程に入る前に製品に対するレビューの強化を実施することができ、品質向上を図ることができる。 ・未決事項を共有することができ、解決に向けた動きがスムーズに行うことができる。		○	○	○	○	
				④結合テスト、システムテストのケース密度・障害密度の基準値を定め、品質評価を行う。 **●工夫・対応の効果** 障害発生率が高い場合、ユーザは追加テストを要求することができ、製品品質の向上を図ることができる。		○	○	○	○	
		上位設計書に対する下位設計書(またはプログラム、テスト仕様)の必要十分性(トレーサビリティ)検証	9－①	①仕様書と試験仕様書、概要設計書と詳細設計書に、トレーサビリティマトリクスを活用。さらに網羅性検証のために、チェックシートに基づく要件定義の具体化を確認・検証するツールを導入し実現する。 **●工夫・対応の効果** ツールを使用することで実現工数はあまり要していないが、トレーサビリティが確保されるため機能の漏れがなくなり高品質の製品提供が実現できる。		○	○		○	
				②上位設計書と下位設計書の記述を相互に記述項目の番号で対応付けることにより、必要十分性を検証する。設計書とプログラムおよび設計書とテスト仕様書の必要十分性検証方式は課題である。 **●工夫・対応の効果** トレーサビリティが確保されるため、機能の漏れがなくなり高品質の製品提供が実現できる。		○	○		○	

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
機能性	正確性 ソフトウェア仕様(又はシステム仕様)とソフトウェア実装(又はシステム実装)との合致度合いを高めるために講じる事例	上位設計書に対する下位設計書(またはプログラム、テスト仕様)の必要十分性(トレーサビリティ)検証		③チェックリストを使用し、漏れがないか確認することで、トレーサビリティを保証する。 **●工夫・対応の効果** プロジェクト参加者がトレーサビリティを常に意識した開発を行うことにより、結果として機能の漏れが減少する。		○	○		○	
				③下位設計書のレビューはその上位設計書をインプットとする。設計書(のコピー)上に試験項目をプロットすることにより設計内容と試験項目の整合性を確保する。 **●工夫・対応の効果** レビュー工数は増加するが、プロジェクト参加者がトレーサビリティを常に意識した開発を行うことにより、結果として機能の漏れが減少する。			○		○	
				④システム性能は各工程で目標値との乖離にて評価する(開発期間中に機能が追加されたり、処理方式の変更、前提条件の変化が発生する。一つ一つの変更の影響は小さくてもそれらが集積されると性能に無視できない影響が発生する)。個々の変更点の影響を評価するだけでなく、各工程でマクロに評価を実施する。 **●工夫・対応の効果** 変更を取り込む際に性能要求に対する影響を考慮すること、また、最終的に確定した機能により性能要件を見直すことで、性能に対する意識が高まり、結果として当初想定性能からの逸脱を抑える。		○	○		○	
				⑤上位設計ドキュメントからプログラムソースコードを自動生成する領域を拡大する。 **●工夫・対応の効果** 設計ドキュメントと実装したソースコードは、仕様変更やデバッグにより実態として乖離が発生するが、自動生成によりドキュメントからコードを生成することで、この乖離を防止する(生成コードには一切手を加えない。ドキュメントを修正することでコードを変更する)。					○	
	相互運用性 同一システム内のプログラム間や他のシステムのソフトウェアとの間で、データやコマンド等をやりとりできる度合いに関する要求水準を保全するために講じる事例	外部システムとの接続要求および仕様の確認(レビュー、テスト)に対するユーザとベンダとの合意		①他システム接続時には障害発生時の責任分担を明確にすることが必要であるため、下記項目に関し事前にユーザ、外部システム担当会社と合意を得る。 ・障害対応体制。 ・境界点。 ・取得情報。 ・意思決定手段。 **●工夫・対応の効果** 設計段階で責任分担を明確にするため、テスト実施時、運用開始時の体制構築がスムーズに行える。			○			
				②外部システムの外部インタフェース一覧、および、インタフェース仕様書をユーザ・ベンダが共同でレビューを実施。ベンダサイドから上がった質疑については、ユーザが取りまとめ、外部システムサイドへ問い合わせを行う。 **●工夫・対応の効果** 設計段階から外部システムの開発ベンダと仕様確認を行うため、インタフェースの誤認識、ミスが減少する。			○			
			5－② 10－①	③外部接続時は、プロトタイプを作成し、早い段階で疎通確認およびインタフェース確認を行う。 **●工夫・対応の効果** 外部接続での課題を早期に発見することが出来るため、他開発に影響を与えることが少なく、コストダウンにも寄与する。		○	○		○	
				④外部システムはベンダからは見えない部分であるが、同時にユーザも正確に把握していることが少ないため、合意事項として外部システムの洗い出しを行い、接続設計に関しては外部システムの開発ベンダも含めて、インタフェース仕様を確認する。 **●工夫・対応の効果** 設計段階から外部システムの開発ベンダと仕様確認を行うため、インタフェースの誤認識、ミスが減少する。			○			
			11－①	⑤外部システムとの接続仕様確認では、相手側のシステムの理解度不足や組織間での文化、標準化の違いにより仕様漏れを引き起こす可能性が高い。以下の対策を実施することで、レビューおよびテスト品質を向上させる。 a) レビュー時には双方の有識者を参加させ、双方のシステムで仕様の妥当性を検証する。 b) 接続双方で洗い出したテストケースは統合せず、個別ケースとして実施する(ケースの統合により双方のシステム固有の確認項目が欠落してしまうことを回避する)。 **●工夫・対応の効果** 双方の観点で確認するため、インタフェースの誤認識、ミスが減少する。またテスト漏れによる障害発生を事前に防止することが出来る。			○		○	
		外部接続に対するデータ数、変換、編集(データ形式、コードなど)に関するコミュニケーション		①汎用系とオープン系とのデータ接続では文字コード系の障害がよく起きる。特に、MIXフィールド(全角/半角混在)については、領域の途中で区切るとシフトコードを付け替えなど煩雑な対応になるため、障害の対処に相当の期間とコストを要する。本課題に対応するため、データ変換機能を共通化する。 **●工夫・対応の効果** より効率的にテスト検証や障害対応を行うことができる。					○	
			11－②	②外部システムのコミュニケーションミス(確認漏れ)の防止、および障害発生時に迅速なリカバリ処理を実現すべく、以下の対策を実施する。 ・外部インタフェース確認内容(データ仕様、受渡時期、順序、編集方法等)をチェックリスト化し、設計工程時に接続双方で突合せチェックを実施、認識齟齬の早期発見に努める。 ・障害発生時の対応について、役割および手順を整理(コンティンジェンシープランの作成)し、利害関係者の合意を得る。 ・迅速なリカバリ処理を実現するため、コンティンジェンシープランの各ケースに合わせ接続双方の回復手順を詳細化し、テスト時に訓練を実施する。 **●工夫・対応の効果** 確認内容をチェックリスト化することにより、不明点を早期に解決することができる。またリカバリ処理に対する認識あわせ、テスト実施により障害時の早期回復が期待できる。			○			○
				③社外とのデータ連携では、通信環境や業務プロトコル等の設定パラメータ量が多く、また必須やオプションの混在もあり、設定内容に関する相互の認識ズレによる設定ミスが生じる。このため設定パラメータに認識齟齬を生じないよう相互接続確認書等の環境設定申請様式や記載事項を明確にするなど見直しを行う。 **●工夫・対応の効果** インタフェースの誤認識、作業ミスが減少する。			○		○	
				④既存システム更改においては、既存システム内に存在する不正データにより、移行作業に多大な労力を要する場合がある。本課題の対処として、移行作業前に現行データの内容を確認し、想定外のデータが存在する場合は作成経路を明確にし、当該データの対応方法についてユーザと合意を得ておく。 **●工夫・対応の効果** 当該対応により、効率のよい移行作業ができ、不正データによる障害発生を未然に防止することができる。			○	○		○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
機能性	相互運用性 同一システム内のプログラム間や他のシステムのソフトウェアとの間で、データやコマンド等をやりとりできる度合いに関する要求水準を保全するために講じる事例	外部接続に対するデータ数、変換、編集(データ形式、コードなど)に関するコミュニケーション		⑤外部接続先との認識齟齬がサービスインの直前に発覚する場合、サービスインの延期を余儀なくされるケースが多い。(例:パンチ会社へ依頼したデータレイアウトに齟齬あり。直前のパンチテスト実施にて気づいたため、サービスインを延期)外部と連携する箇所においては、すべて早期に対応を行い、インタフェースの誤認識、ミスを排除しておく。 **●工夫・対応の効果** 早期に障害を発見することで、納期に対する影響を抑止することができる。			○		○	
				⑥国内システムと海外システムとの連携において、想定外の事象が発生する場合がある。(例:日本時間から海外への日時変換にて、サマータイムの切替り時期で日が変わることがあり、結果同一日にふたつのデータが存在するという障害が発生)想定外に発生した障害については、以降の開発において同様の誤りを発生させないよう、チェックリスト化する。 **●工夫・対応の効果** 類似障害の発生を未然に防止することができる。			○		○	
		相互接続する他のソフトウェアのバージョンアップによる影響に対する考慮	12−①	①ミドルウェア製品のバージョン変動時(アップ/マイナーアップ)には、そのミドルウェアを前提とした製品の非互換を確認する。同一環境にあるソフトウェアのバージョンアップ情報を関連箇所に通知する仕組みを構築し、独断でのリリースは行わない。 **●工夫・対応の効果** バージョンUPする内容により影響を受ける範囲を特定できるため、ユーザとの調整、合意形成が容易となり、早期の対策と取ることができる。			○	○		
			12−①	②相互接続するソフトウェアのバージョンアップ時は必ず情報連携し、接続インタフェースの見直しを行い同期を取ってバージョンアップする。 **●工夫・対応の効果** バージョンUPする内容により影響を受ける範囲を特定できるため、ユーザとの調整、合意形成が容易となり、早期の対策と取ることができる。			○	○		
				③過去開発でのソフトウェアバージョンにかかわる事例(例:インターネット関連システムにおいて、同一サーバ上搭載された証明書検証関連ルーチン(他部門開発モジュール)と、自部門アプリケーションがそれぞれ利用しているJavaのバージョンの不一致が許されないケース)で得たノウハウをチェックリストとして部門単位に蓄積する。 **●工夫・対応の効果** ノウハウを活用することにより、早期な対策(上記の例では別なサーバでの開発をユーザに依頼する等)を取ることが可能となる。			○	○		
		接続先外部システムとのコンティンジェンシープラン共有化		①関連する業務システム双方に改修が発生している場合、開発スケジュールのずれによりシステムにて業務を実現できないケース(エンドユーザによる手作業発生)が起こりえる。本課題に対し、以下の対応を行う。 ・関連するシステム開発が平行で行われている場合、連携部分のリリース時期をユーザと合意しておき、進捗状況を定例でユーザから確認する仕組みを構築する。 ・同時開発のプロジェクトに変更が入るなどスケジュールに影響を及ぼす要因が発生した際に、情報が収集できる体制を事前に構築する。 ・同時開発のプロジェクトのリリースが遅れることが想定される場合、対応策についてユーザと事前に合意する。 **●工夫・対応の効果** 事前に影響範囲や想定外作業の有無を明確にすることで、あらかじめ作業計画が立案でき、障害発生を未然に防止することができる。						○
			13−⑤	②開発を同時期に行う接続外部システムが存在する場合、コンティンジェンシープラン作成時はすべての開発ベンダが参加し作成することにより、対応策、担当範囲を明確にする。多数の利害関係者が存在する場合、他者の進捗状況により影響を受けることが予想されるため、可能であれば、要件定義の段階より他者を巻き込んだ形でプロジェクトを進めていく。必要により、全体を統括するプロジェクト管理チーム(ユーザ主体)を設けることも検討し、提言する。 **●工夫・対応の効果** ・すべての利害関係者が対応策について合意することにより、早急な対応を行うことができる。			○	○		○
			13−⑤	③汎用機/オープン機が連携してしている場合にオープン機がダウンしていると、取引に関する全てのサービス提供が停止するような構成になっていることがある。取引の全面停止を避けるために、オープン機がダウンしていても、取引の予約だけは出来るようにした。 **●工夫・対応の効果** 取引に関する全てのサービス提供が停止すると、ビジネス上の損害は多大になる。当該対策を実装することで、当該リスクを低減できる。			○	○		○
	セキュリティ データ、システムまたはソフトウェアへの不正なアクセスを排除する機能や、ミスによる資源破壊の防止の度合いの要求水準を保全するために講じる事例	データ暗号化対策	14−①	①特に高い機密性が要求されるシステムにおいて、パスワードの暗号化を2段階にする。 基盤(DBI)で暗号化する上に、更に各業務で暗号化する。 **●工夫・対応の効果** 当該対応のためのコストはそれほど大きくないが、両方の暗号化ルールを知っている人は殆ど存在しないので、パスワードの漏えいリスクの低減に寄与する。		○				○
				②社外持ち出す出向業務用携帯端末に対して、通信ポート保護、格納データファイルの暗号化やユーザ認証に加え、紛失及び盗難時には利用状況を「紛失」に変更することでオンライン業務接続を抑止することができる等のセキュリティ実装を行う。 **●工夫・対応の効果** 情報漏えいリスク低減に寄与する。						○
				③各種暗号化すべき対応項目を整理する。 ・データベースに格納されたデータの暗号化。 ・通信の暗号化:VPN(IPsec)、SSL(TLS)等。 ・端末暗号化。 ・原本性(真正性)確認レベル:時刻認証(タイムスタンプ)、署名(ハッシュも含む)、印刷データ電子化・管理など。 **●工夫・対応の効果** 暗号化するためにはコストがかかるため、必要以上の暗号化を用いるべきではない。外部からの脅威に晒されるかどうかデータごとに判断し、暗号化の要否を検討すべきである(コストと漏洩リスクのトレードオフ)。		○				○
		開発時でのデータ漏洩防止対策		①障害解析や、テストのために、よく本番データを使用する。この場合に、機密性を保全するために、本番データから、機密情報や個人情報(氏名・住所・電話番号等)を暗号化して抜き出す機能を実装した。 実装方法は、下記に示すとおりいくつかの形式がある ・ファイルごとに、データマスクをする開始位置・桁数・データ種類(氏名等)を定義する形で汎用化を図っているもの。 ・開発部門でマスキングジョブを予め準備し、運用部門に依頼してマスク済のデータを提供するもの。 **●工夫・対応の効果** ・本番データの使用頻度は高いため、使用の都度対象の項目を暗号化するのに要するコストを、当該機能を利用することによって、大幅に削減できる。 ・機密情報にマスクをかけることにより、漏洩リスクが低減できる。 ・本番データを利用することにより、テストデータ作成工数も削減できる。	○			○	○	○
			14−②	②情報セキュリティ管理策(本番データの取り扱いルール、個人情報等機密データのマスキング、持ち出し時のデータ所管部署の許可、証跡の確保等)を構築する。 **●工夫・対応の効果** ・機密情報にマスクをかけることにより、漏洩リスクが低減できる。 ・本番データを利用することにより、テストデータ作成工数も削減できる。	○	○				○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
機能性	セキュリティ データ、システムまたはソフトウェアへの不正なアクセスを排除する機能や、ミスによる資源破壊の防止の度合いの要求水準を保全するために講じる事例	セキュリティ事故発生に備えた、追跡可能性および監査容易性などの向上施策		①開発者が本番データを参照・編集できるIDは、運用部門への事前申請を必要とする。使用するデータセット名を予め申請し、事後にログデータと突き合わせて確認を行う。(アプリ基盤に監査証跡を記録するためのログ機能を装備) **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。			○	○		○
				②開発端末からのデータ漏洩には以下の対策を行う。 ・外部媒体書き出しは、上司への申請を必須とし、ログデータと付きあわせする。 ・インターネットメールによるデータ伝送はパスワードを必須とし、メール履歴から上司がチェックを行う。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
				③PC付属の外部記憶媒体の無機能化を実現する。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。			○	○		○
				④PC付属の外部記憶媒体(コネクタ)に外部記憶装置を接続するだけで、監視システムにアラームがあがるような仕組みを構築する。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。			○	○		○
			15－①	⑤特に機密性の高いシステムの場合、以下のような対策を実施する。 ・不正アクセスを検知する仕組み:不正な通過パケットを自動的に発見するIDS／IPS等の措置を講じる。 ・重要な情報設備の、他の区画との明確な区別。 ・入退室記録の保存。 ・監視カメラの稼働、監視映像保存。 ・警備員の常駐。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
			14－③	②システム毎に異なっていたセキュリティログ(ID、資源アクセス記録etc)を定義、ログフォーマットを標準化、運用等社内ルールを規程。セキュリティログ取得保管のための運用管理支援システムを構築しセキュリティログの一元的な保管・管理を行う(各部門サーバのセキュリティログを定期的に取得し保管する)。 **●工夫・対応の効果** バラバラであったログ情報を統一的に管理することによる、進入経路や手段等の特定作業が向上する。						○
		セキュリティパッチなど脆弱性の予防対策		①Windowsセキュリティパッチについては、専門の部門にて適用の要不要の判断と共に、関係先へ周知する。 **●工夫・対応の効果** セキュリティホールの排除に寄与する。						○
				②開発用PCにインストールされているソフトウェアを運用管理者が監視し、許可されたものかチェックしている。 **●工夫・対応の効果** セキュリティホールの排除、情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
				③セキュリティパッチは自動送信することにより、全PCの状態を正常に保つ。 **●工夫・対応の効果** セキュリティホールの排除に寄与する。						○
				④Webアプリケーションの改定・開発を行う場合はWebアプリケーション脆弱性評価ツールによる診断を実施し、定型の診断報告書をサービスインレビュー時に確認する。 ルール化の前に、既存アプリケーション全てについて診断を行い、改定時の診断の負荷を軽減する。 **●工夫・対応の効果** 情報漏えいリスクに寄与する。						○
				⑤開発テスト工程で検証ツールにて開発システムのセキュリティ診断テストを実施し、本番機においても第三者によるペネトレーションテスト(セキュリティ診断)を実施後に運用開始を定める。 またITILベースの構成管理支援ツールを導入し、各部署に散在していたH／W、S／W管理情報を一元化する。 **●工夫・対応の効果** セキュリティホールの排除に寄与する。また類似障害対応機種やパッチ適用機種の絞込み作業を迅速に、かつ的確に実施することができる。				○	○	○
				⑥システムの脆弱性を排除するため、下記対応(工夫)を行う。 ・セキュリティリスク管理:セキュリティリスク分析、セキュリティリスク対策、セキュリティパッチ適用。 ・ネットワーク診断:ネットワーク等への外部からの侵入に関する脆弱性の診断(外部委託を含む)。 ・Web診断:Webアプリケーションサーバ等への外部からの侵入に関する脆弱性の診断(ポートスキャン等)。 ・DB診断:DB等への外部からの侵入に関する脆弱性の診断(外部委託を含む)。 **●工夫・対応の効果** セキュリティホールの排除、情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
				⑦(保守運用) ・必要に応じて随時適用(原則適用しない)						○
		不正アクセス、不正ログインに備えたアクセス制御対策	16－①	①SSOによる統一的な認証基盤を構築。IDパスワード変更管理、誤入力とアカウントロック等の社内セキュリティポリシーを実装し運用する。社外からのアクセスユーザはICセキュリティカードを発行。カード申請、発行、更新および失効等の管理業務システムを構築し運用管理を実施する。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
				②ログイン時に前回ログインした日時を画面に表示することにより、他人に使われた事(不正ログイン)を早期に発見する。 **●工夫・対応の効果** 不正アクセスの防止に寄与する。						○
				③離席時に、PCをロックすることを徹底。また、自動ロックの時間を5分にするよう運用を徹底する。 **●工夫・対応の効果** 不正アクセスやのぞき見の防止に寄与する。						○
				④PCにIDカード(社員証)を挿入することにより、ユーザの使用資格(使用可能画面)を判別する仕組みを構築する。 **●工夫・対応の効果** 担当業務以外への誤操作防止、情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
				⑤本番データへのアクセスログをチェックすることで不正アクセスを検出する。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
				⑥ユーザIDは個人にアサインし、複数人による共有を禁止する。パスワードは、定期的な変更を強制する。 **●工夫・対応の効果** 不正アクセスの防止に寄与する。						○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
機能性	セキュリティ データ、システムまたはソフトウェアへの不正なアクセスを排除する機能や、ミスによる資源破壊の防止の度合いの要求水準を保全するために講じる事例	不正アクセス、不正ログインに備えたアクセス制御対策	16－①	⑦セキュリティプログラミングのガイドライン(社内で用意しているセキュリティを確保するための留意点等をまとめたガイドライン)を活用する。(お客様から預かった業務資産(アプリ、データ)について、業務ごとにアクセスできる人間を制限し、開発環境とアカウントを分離する等)また、本番環境(本番稼働サーバなど)にアクセスできる端末は利用申請制でかつ監視室内設置の専用端末とする。アカウントも申請毎に変更する。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。		○	○	○		○
				⑧不正アクセス防止のため、システムごとに、下記の例に示すような認証管理を実施する。 ・管理者のアクセス認証管理:デジタル証明書による認証、生体認証、ICカード、ID・パスワード ・利用者のアクセス認証管理:生体認証、ICカード、ID・パスワード **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。						○
		なりすましのモニタリング、警告機能などに関する工夫	16－②	①不正侵入検知システム(IPS／IDS)を導入する。 **●工夫・対応の効果** 情報漏えいリスクの低減、不正アクセスの防止に寄与する。		○				○
		データ損傷などのデータ保全対策		①バッチ処理の最後にDB毎の「整合性検証」を組み込んで、DB破壊(不正コード、キーアンマッチなど)のエラー摘発を行う。相当な規模になるため、DBが保持する情報の重要性を評価して「整合性検証」対象のDBを絞り込む。				○		○
			16－③	②システム構成上、物理配置上の工夫として下記を行う。 ・ファイルの二重化。 ・バックアップセンタへのログおよびファイル転送(バックアップセンタのデータベースはメインセンタから送信されたログを元に同期あわせを行っているが、定期的に、メインセンタとバックアップセンタそれぞれで同期完了後の状態で取得した当該データベースのバックアップファイルをコンペアして整合性を確認している)。 ・バックアップ(正・副)の取得。 ・ストレージの冗長化。 ・ストレージは、SANにしている。 ・データのバックアップ。 ・バックアップデータの保管場所分散。 **●工夫・対応の効果** ・ディスクの破損に対する復旧時間の短縮が図れる。 ・局所的な自然災害の場合、早期の復旧が可能となる。			○		○		○
				③高信頼度ディスクを導入、各システムで共用利用する。 ディスクボリュームを正副に分割。正ボリュームはアクセス性能を重視し、RAIDグループをまたがったストライピングを実施して業務データを格納し、副ボリュームは耐障害性を考慮し単一RAIDグループ配置とし、バックアップ用また非常災害時の遠隔地保管用レプリカとして設計する。 **●工夫・対応の効果** ・ハイエンドストレージシステムを各業務システムで共用し、また業務毎に発生していたバックアップやリストア等の運用方式を標準化することにより設計・運用業務の効率化が図れる。 ・ディスク容量計画をスモールスタートと定期的使用率把握をしつつ増強する方針に改め、ディスクの余剰領域を有効的の活用できTCO削減が実施できる。 ・ストレージシステムの設計においても正・副の量ボリュームの使用特性に応じたディスク量を見積もりコスト抑制を実施した。 以上コストを抑えつつデータ保護に対する信頼性が向上した。		○				○
		セキュリティ要件(実装検証含む)のユーザとベンダーとの合意	15－②	①ISO/IEC15408)[CC(Common Criteria)/ST(Security Target)]認証レベルで合意し、第三者機関によるセキュリティ設計仕様書(ST)のST評価・ST確認を実施する。 **●工夫・対応の効果** セキュリティチェックを第三者が行うことにより、システムの脆弱性を客観的に評価することができるとともに、攻撃対象と思える箇所に対し、事前にリスク軽減の対策を施すことが可能となる。	○					
				②実績があり、有効度が認識されているアタックテストや、コードの書き方のチェックツールを用いて納品物チェックを実施することを契約条件とする。 **●工夫・対応の効果** セキュリティチェックを行うことにより、システムの脆弱性を客観的に評価することができるとともに、攻撃対象と思える箇所に対し、事前にリスク軽減の対策を施すことが可能となる。	○			○		
	機能性 標準適合性	機能に対応する規定(業務、内部統制、ISMS、国際会計など)および開発標準の適合監査対策		①要件定義作業で明確にすべき項目とそれらの関係が明確に記述されているか、またその内容はビジネス要求から見て適切か、などを監査することを実施する。(ただし、第三者で検証できる形式チェックのみ) **●工夫・対応の効果** 形式チェックではあるが、第三者による内容監査を実施することで、法令・規格・基準・ガイドラインの遵守状況を客観的に評価できる。		○				
				②海外システムにて、国際基準との整合性につき外国の専門家による設計方針レビューを実施する。 **●工夫・対応の効果** 国際法との整合性を保つことができ、問題の発生を未然に防止できる。						○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性（2次）	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	成熟性 ソフトウェアのフォールト[Fault]により発生する故障[Failure]の頻度に関する要求水準を保全するために講じる事例	潜在的障害予測（又は潜在的障害低減）および障害解決状況の分析	17－①	①下記の手法を用いて、残存不良を推定し製品品質、信頼性を客観的に評価する。 ・サンプリングテスト：QA部門により検査項目の10%を目安に試験を行い、摘出された不良件数から残存不良を推定する手法。 ・信頼度成長曲線モデルによる推定：テスト工程の80%時点を目安に、摘出された不良件数からモデルをあてはめ残存不良件数や信頼度、MTBFを推定する手法。 ・ゴンペルツ曲線による推定：変曲点より総バグ数を求めることにより、バグ収束状況を判定する。 **●工夫・対応の効果** 製品の信頼性を客観的に評価することができる。			○		○	
				②プロジェクト運用に入る前に、障害発生目標値を設定し実績を収集するとともに、定期的に分析、評価を行う。過去のプロジェクトの障害率を参考に、テスト工程の障害率の目安を設け、サービスインレビュー時に障害件数を評価する。 **●工夫・対応の効果** ・類似プロジェクトの障害発生に関する指標を利用することで、製品品質、信頼性を予測することができる。 ・障害件数の過不足が顕著で信頼性に欠ける場合、追加テストの実施、サービスインの延期等について検討することができる。					○	○
			17－①	③システムテスト、運用等にて発生した障害については、事象、原因、対策、類似機能の有無を障害連絡報告に記載するとともに、再発防止策を立案する。類似機能が存在する場合、調査範囲を特定し、ユーザと合意を得た上で調査、場合により改修を実施する。障害を迷惑度別に切り分け、迷惑度の高い障害については、関連する各社の役員、本部長クラスが出席する会議で報告、検討することにより、障害原因、対策の周知を図る。 **●工夫・対応の効果** ・障害発生時に関連機能を調査することで、更なる障害の発生を未然に防止する。 ・障害の原因となった事項をチェックリストに追加し、類似する障害の発生を未然に防止する。			○	○	○	○
		仕様変更管理（傾向、内容分析）に基づく、仕様変更の制御	4－②	①仕様変更に関する発生、回答の推移をモニタリングするとともに、仕様変更内容を分析し、内容・観点別分類しチェックリスト化する。 **●工夫・対応の効果** 仕様変更を分析し、横展開を図ることにより、類似機能に関し要件の変更が必要か判断し、後工程での手戻り工数の発生を抑止する。		○	○	○	○	
			4－②	②仕様変更に顕著な傾向が見られる場合、対応策を検討する。対応策が解決するまで、開発を一時中断することも併せて検討する。 **●工夫・対応の効果** 仕様変更を分析し、偏った機能あるいはサブシステムに仕様変更が頻発している場合、当該機能・サブシステムについて再度ユーザと機能調整、合意する仕組みを持つことで、後工程での仕様変更の発生を抑止する。			○			
		MTBF向上施策	30－③	①ネットワークの障害に対し、冗長度を高めることにより通信関係の信頼性を高める。異経路の複数回線（異なる通信会社を使用した回線の2重化）を使用する。さらに、バックアップとして携帯電話での通信、FAX、近隣事業所へのデータ持ち込みを可能とする。 **●工夫・対応の効果** ・通信回線の代替ルートを用意しておくことにより、システム全体の信頼性が向上する。 ・通信設備作業による停止を未然に防ぐことができる。		○				○
			18－① 30－③	②サーバ障害に対し、冗長度を高めることにより信頼性を高める。 ・DBを持たないWeb／アプリケーションサーバなどはスケールアウト構成、DBサーバはクラスタ構成。 ・待機系サーバ構成によるサーバ障害時の切替による継続運用。 ・負荷分散装置の適用による複数サーバ構成の運用。 **●工夫・対応の効果** サーバ障害時の代替機を用意しておくことにより、システム全体の信頼性が向上する。		○		○	○	○
				③システム構成要素におけるシングルポイントの局所化する（基本的にハードウェア、基本ソフトウェアの機能による）。 **●工夫・対応の効果** シングルポイントの局所化を図ることにより、可用性の向上を図るとともに、障害発生時の復旧時間の短縮が実現できる。		○				○
				④MTBF＝サービス稼働率の定義について、お客様と事前に合意する（実際に行った対策を以下に記載）。 ・障害時のサービス継続性を確保できる設計（各種方式によるシステムの冗長構成）。 ・オペミスを防止するために自動化処理組み込み。 ・アクセス頻度を抑制するセッション管理。 ・高信頼化ミドルウェアの開発。 ・システム稼働状況やディスク容量等の監視・通報機能の動作。		○	○	○	○	○
				⑤開発プロセスの型決め（標準化）ソリューションを用意する。 **●工夫・対応の効果** 開発/保守プロセスの効率化と成果物の品質向上に寄与する。				○	○	
				⑥予防保守の機能（自動故障予測検知）整備、強化によりトラブルを事前に防止する。 **●工夫・対応の効果** システム異常が早期に発見できるため、システム異常時の影響拡大を未然に防止する。						○
		「テストコスト」と「本番稼働後の欠陥による社会的影響や復旧にかかるコスト」とのトレードオフを考慮したテスト手法（テスト網羅率）の取り込み		①脆弱性を抱えるWebアプリケーションは、Webブラウザから容易に攻撃を行うことができ、かつ、被害が発生した場合のインパクトが非常に大きいため、Webアプリケーションをリリースする前に十分に脆弱性の検証を行う。 **●工夫・対応の効果** ・ツール使用により、セキュリティ脆弱性検査の効率性、網羅性向上。 検査対象を設定することで様々な攻撃手法に対応した大量の検査項目が自動実行できる。またアプリケーション修正後の検査の反復が容易である。 ・セキュリティ専門体制確立により、効果的、効率的な検査・対応を実施できる。					○	
			8－①	②システムの各機能（画面、帳票、ファイル・インタフェース）について、障害が発生したときに発生するコスト（機会損失なども含む）とテスト項目網羅率に基づくテストコストの比較表を作成し、機能ごとにテストの軽重を決定しテスト戦略に反映する。機能ごとに本番障害による社会的影響度合い（金額）を評価し、目標のテスト密度を設定する。開発コスト削減が必要な場合でも当該影響度が高い機能については十分なテストを実施する（＝テストコストは落とさない）。 **●工夫・対応の効果** 機能ごとに評価することで、効果的なテスト実施、開発コストのコントロールが可能となる。					○	
			2－③ 35－③	③本番稼働後のリスクを勘案した試験項目を抽出し、実施する（一部ではリスクシートに表現）。リスクに応じテスト実施要否を判定する。 **●工夫・対応の効果** 機能ごとに評価することで、効果的なテスト実施、開発コストのコントロールが可能となる。					○	
				④（開発）テスト範囲・深さを拡大することとコストの制限を両立するため、単体テストの段階からテストの仕方を決めてツールを導入し、リグレッションテストなどの自動化を行う（テスティングプロセスの型決め（標準化）ソリューション）。 **●工夫・対応の効果** ・テスト工程を標準化することにより、品質の均一化、および、コスト削減を実現できる。 ・障害発生時は、類似処理、類似テスト状況を確認することにより障害の拡大を未然に防止する。					○	

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	障害許容性(発生防止策) ソフトウェア(又はシステム)のフォールトや予期しない入力があった場合でも、重大な障害や中断を引き起こさず、表面上問題なく動作することができる度合いに関する要求水準(発生防止)を保全するために講じる事例	イレギュラー処理の実装に関する工夫	18－②	①DBへの大量アクセスによるバッファ不足に対し、一定件数ごとにバッファを開放するようなDB検索方式を採用する。 **●工夫・対応の効果** オンライン停止障害発生を未然に防止できる。			○			○
				②ユーザへのサービスが停止するという障害を防ぐため、オンライン機能には異常終了処理にてオンライン停止処理を組み込まないようチェックリストに追加し、運用する。 **●工夫・対応の効果** オンライン停止によるユーザ業務への影響を未然に防止できる。			○			○
				③バッチ処理に警告用に組み込むコンソールメッセージについては、多発により運用管理者よりクレームが発生する可能性があるため、上位設計者と発生頻度等考慮し組み込むようチェックリストに追加する。 **●工夫・対応の効果** 日々の運用業務への影響を必要最小限にとどめることができる。			○			○
				④運用手順の考慮漏れは対応遅延(または誤り)に繋がるため、運用手順について、ユーザの運用部門も含めたレビューを実施する **●工夫・対応の効果** レビュー実施により、運用手順の妥当性や正確性を評価でき、作業全体の品質向上が図れる。						○
		障害を事前に検知するためのサブシステム間のトレース機能の工夫	17－②	①サブシステム間のシステム整合性チェック機能を準備し、不整合発生時は注意喚起する仕組みをとる。 本機能は開発環境(システムテスト)にて他サブシステムも検証用として利用する。 **●工夫・対応の効果** システム全体の不整合の早期発見に寄与する。			○	○		
				②ログファイルの配置について、以下留意する。 ・オンライン処理とバッチ処理でログファイルを分離する。 ・ログファイルは日次でバックアップする運用とする。 **●工夫・対応の効果** オンライン処理とバッチ処理にてログファイルを共有するような仕組みの場合、バッチ処理の実施有無によりオンライン処理時点の情報が消去されている可能性がある。上記対応により、必要なログファイルを確実に取得することができ、障害原因の特定に寄与する。			○	○		
		障害に対応するためのテスト環境準備に関する工夫		①本番ミニDBを構築する仕組みを実装する。 **●工夫・対応の効果** 本番DB全件はデータ量的に無理なので、キー(障害が発生した営業店など、いくつかの抽出パターンを用意して)を指定して、ミニDBを構築するバッチ処理を構築する。 工夫・対応の効果 本番で使用しているデータでテストを行うことができるため、機能の考慮漏れが減少する。					○	
			20－①	②特に重要インフラ等システムでは、本番環境でのテストケースを実施(テスト密度不足への対応)すべく、本番と同等性能の障害テスト確認環境(24時間利用可能)を構築し、障害対応時に利用する。 ・データは、個人情報に配慮したマスク済の本番データを利用する。 ・本番環境での確認用に、テスト用口座やテスト用ユーザIDを設ける **●工夫・対応の効果** 本番同等の環境でテストを行うことができるため、機能の考慮漏れが減少する。					○	
			20－①	③本番使用JCLをテスト機用に自動変換する仕組みを実装し、かつ、個人情報をマスクした本番データの使用を可能とすることで、テスト環境で本番環境とほぼ同一のテスト(再確認)を実施可能にする。 抽出パターンを指定することによりテストに必要な最低限のデータのみ抜き出すことを可能とする。 **●工夫・対応の効果** ・本番JCLをテスト環境向けに自動変換するため、テスト環境の構築が容易に行え、障害の解析およびテスト確認コストも削減できる。 ・本番JCL作成後にテスト環境向けにJCLを作成するため、本番JCLの確認が行え、品質向上に繋がる。					○	
				④新規システム構築のためサーバーを新規購入するような場合で、本番使用サーバーを試験用に流用する前提のもと、開発環境に関する調整が疎かになるケースが存在する。当該ケースに備え、性能の劣るサーバーを別途用意する等、開発環境に対しユーザとの事前合意を行う。 **●工夫・対応の効果** 開発環境と本番環境の差異を明確にすることで、環境相違による認識齟齬、品質の確認漏れなどを未然に防止することができる。					○	
			37－③	⑤業務の準本番環境(保守環境)を構築し、最終検証をこの環境で実施する。 **●工夫・対応の効果** 本番同等の環境でテストを行うことができるため、機能の考慮漏れが減少する。					○	
				⑥通常は分割して使用しているが、構成変更により本番システムとほぼ同等な環境が構成できるだけの資源を準備する。アプリケーション、JCLが意識する環境(ファイル名、ファイルサイズ等)は、本番と開発環境で同じ構成とする(環境を意識して変更を加える必要がない)。 **●工夫・対応の効果** 本番同等の環境でテストを行うことができるため、環境構築の手間が省けるとともに、機能の考慮漏れが減少する。					○	
				⑦ホスト系オンラインシステムの開発環境では、本番プログラムライブラリと、開発プログラムライブラリを同時に使用できるようにし、動作や処理結果の比較を容易にする。 **●工夫・対応の効果** テスト結果の新旧比較が容易に行え、テストの効率化を図れる。					○	
				⑧UNIT等を用いたテスト駆動開発の実施および実施後のテストコードの活用 ・障害発生後、本番環境に修正モジュールを適用する際に実施するテストのうち、第一ステップで実施するテストコードを活用した試験、および、出荷試験項目表(ソフトウェア修正の際には、リリース前に必ず実施する最低限の試験項目)を準備する。 ・サイト毎に異なる環境・データが必要な場合、メニュー起動でターゲットサイトに切り替わる仕組みの構築する。 **●工夫・対応の効果** ・可能な限りの自動化を図ることで、人間系介在により誤りを削減できる。 ・出荷試験項目表を作成することで、以降の改定時の機能理解が図れ、工数削減も見込むことができる。					○	
			19－①	⑨運用開始に向け、移行計画(リハーサルを含む)について事前準備を行う。 a)計画範囲: データ移行、関連サーバ切替;N／W環境切替、端末切替、業務切替 b)計画内容: 【手順整備】移行手順(DB展開時間etc)・検証手順の取り決め、 【役割整備】移行作業時の要員配置・役割分担の取り決めと作業環境の把握、 【不測事態への対策】発生時の役割分担の明確化、情報連絡ルートの確立、戻し手順 【教育】移行作業における要員教育、 【リハーサル実施】実施日程および回数 **●工夫・対応の効果** 作業手順の検証精度が向上し、所要時間も削減する。					○	

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	障害許容性(発生防止策) ソフトウェア(又はシステム)のフォールトや予期しない入力があった場合でも、重大な障害や中断を引き起こさず、表面上問題なく動作することができる度合いに関する要求水準(発生防止)を保全するために講じる事例	障害管理(傾向、内容分析)に基づく、障害の予防・是正処置の実施	21－③	①過去の類似プロジェクトでの欠陥混入パターンを分析し、同様プロジェクト開発時に、チェックリスト運用を実施する。不良分析(現象、原因、作りこみ原因、見逃し原因)の工夫として、以下の対応を行う。 。なぜなぜ分析(なぜの繰り返し)を行う。 ・4倍返し(検査部署で摘出された不良の4倍の類似不良を摘出)を目標とする。 ・発生した障害の事象および原因から、調査範囲や抽出する観点を導きだし、他の同事象もしくは同一原因に起因する不良がないか調査を行う。 ・障害の誘因については、詳細分析し、再発防止策を実施(開発用ドキュメントや各種チェックポイントへの反映)する。 **●工夫・対応の効果** 障害原因となった項目、コーディングをチェックリストに追加することで、同様障害の発生を未然に防止することができる。			○		○	○
			21－③	②運賃システムにて、以下のキーにて障害分類し強化テストを実施する。 ・機能面:障害発生率の高い取引(払戻系機能など) ・データ面:障害発生率の高い運賃(往復・乗継系運賃など) ・販路面:障害発生率の高い販路(代理店系) **●工夫・対応の効果** 強化テストを実施することで、成果物の品質向上が期待できる。			○		○	○
				③過去に発生した障害の情報を蓄積し、システム毎に発生頻度を調査する。発生頻度の高いシステムは、フォーカスして原因の深掘りや改善策の実施を行う。 **●工夫・対応の効果** 個別システムごとに評価することで、システムの特徴を把握することができ、過去発生障害の未然防止、対象システム固有の課題抽出が可能となる。						○
				④再現しない障害に対する対策として、システム内に情報採取のための(業務用でない)特別な機能を組み込んでいる。 **●工夫・対応の効果** 発生障害の直接原因を突き止めることができ、再発防止に貢献する。						○
		運用時点での障害発生状況(傾向、内容分析)に基づく、障害の予防・是正処置の実施	21－②	①本番にて発生した障害、および、保守案件対応時に発見した潜在障害については、事象、原因を特定するとともに、障害連絡報告を作成し、他の類似事象に関する調査範囲を特定するとともに再発防止策を作成する。 **●工夫・対応の効果** ・発生障害にランクをつけて、ランク毎の内容や傾向分析を実施。特に重大障害は特別なルールを定めて、障害分析～対策、水平展開を行い、再発を防止する。 ・再発防止策を是正処置・予防処置に展開する。						○
				②ITILを導入する。 **●工夫・対応の効果** 標準化された手順で管理、運用しているため、障害発生時速やかに対応できる。						○
		ミスオペレーションの防止に関する工夫	13－①	①取引の重要度に応じて、取引成立に必要なプロセスを3段階に区分して、オンライン操作を実装した。 A. 低ランク:操作者のみで取引成立 B. 中ランク:操作者＋上位権限者の確認入力が必要 C. 高ランク:異例取引(赤残になる取引など)はさらに上位者権限の承認入力が必要 **●工夫・対応の効果** 多数の取引(アプリケーション)で対応が必要なので、全取引に共通機能として提供し、これを業務共通機能に組み込んで、個別のアプリケーションでは意識しないようにする。また取引毎にA. or B. orCを外部テーブルに登録するようにし、チェック条件の変更はプログラム修正を不要とする仕組みとする。 **●工夫・対応の効果** 利用者の責任範囲が明確化され、かつ複数利用者による相互チェック機能も働くため、操作ミス			○	○		
			13－①	②誤操作防止対策として、ペアオペレーションによる多重操作を実装する。 ・担当者申請⇒上席承認等を必須とする。重要取引については、入力者と検証者による2重オペレーションを必須とする。 ・更新ファイル、設定ファイル等重要作業時には、担当者と監視者の2重オペレーションを必須とする。 **●工夫・対応の効果** 責任範囲が明確となり、複数人での相互チェック機能も働くため、操作ミスによる障害の削減に寄与する。			○	○		○
				③システム停止や業務終了といったオペミスが与える影響が大きい業務については、Ctrlキーを押しながらマウスクリックさせる仕組みを構築する。その他の業務の進捗状況を監視し、業務終了やシステム停止を実施してよいタイミングかどうかにより、その他の業務を実施中の場合には、アラーム表示を行う仕組みも構築する。 その他として、下記対応を実施する。 ・範囲チェックによるあり得ない入力の防止、あるいは警告。 ・あらかじめ作業手順と内容を設定して、その手順通りかのチェックを実施。 ・自動化を行い、起動のみを手動操作とする。 **●工夫・対応の効果** 重要な機能については、操作、手順を複雑にすることで、オペレータに注意を喚起することにより、操作ミスによる障害の発生を未然に防止する。				○		
				④システムからオペレータ向けに出力するメッセージの設計基準を設け、必要最低限のものにする。(多量のオペレータ向けメッセージによって、オペレータがメッセージの内容確認に忙殺されることを防ぐ)。 **●工夫・対応の効果** ・メッセージの内容は大半がシステムの動作に影響の無いものであり、設定基準によりメッセージ出力の削減が図れる。結果、オペレータの作業負荷を軽減させることができる。 ・プログラマはエラー処理を実装するときに、メッセージを出力したがる傾向があり、個々のソフトウエアが出力するメッセージは少なくてもシステム全体では大量のメッセージが出力されることに留意すべき。			○			
				⑤業務システムメニュー形式による業務選択、選択系GUIなどにより、入力を出来る限り簡素化する。 **●工夫・対応の効果** オペレータが誤解することがなくなり、操作ミスによる障害の削減に寄与する。						○
				⑥自動化(製品の適用、処理作りこみ)により、運用手順の省略・入力簡素化を行う。 **●工夫・対応の効果** 人手を介さないことにより、入力ミスによる障害の削減に寄与する。						○
				⑦処理別専任体制を構築し、オペレーション手順チェックシートを作成し、運用する。 **●工夫・対応の効果** 経験者による入力が主体となるため、操作ミスによる障害の削減が期待できる						○
				⑧オペレーション自動化製品を導入する。 **●工夫・対応の効果** 人手を介さないことにより、入力が簡素化でき、入力ミスによる障害の削減に寄与する。			○	○		○
				⑨人手のオペレーションは、原則事前に検証された作業指示書を元に実施する。 **●工夫・対応の効果** 作業指示書に従った入力を行うことにより、操作ミスによる障害の削減に寄与する。			○	○		○
				⑩ホストエミュレータ端末において、操作中で特定キーを押下した場合、これを無効にする機能を組み込む(Escキーの押下によりキーボードロックが解除され、直前の画面が再送されることによる異常データ登録を抑止する目的)。 **●工夫・対応の効果** 不要なキー操作を無効にすることにより、誤ったデータを処理することがなくなり、障害発生を未然に防止することに寄与する。			○	○		

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	障害許容性(発生防止策) ソフトウェア(又はシステム)のフォールトや予期しない入力があった場合でも、重大な障害や中断を引き起こさず、表面上問題なく動作することができる度合いに関する要求水準(発生防止)を保全するために講じる事例	システム資源(CPU、ディスクなど)の使用状況監視、警告に関する工夫		①システム利用拡大に伴うデータ量の増加にあたり、以下の対処を実施する。 ・ディスク使用率を監視する仕組みを実装する(100%ディスクを使い切る前にアラームをあげる)。 ・さらなる改善として、事前に削除・移動可能なファイルを登録しておくことで、アラーム発生時に自動的に不要ログ等を削除する仕組みを構築する。 **●工夫・対応の効果** ディスク容量オーバーによる障害を未然に防ぐことができる。						○
			22－①	②データベース表領域使用容量、断片化率等の監視機能を実装したデータベース運用監理システムを開発、表領域使用量の閾値を設定し、閾値オーバー時に統合運用監視システムへの通知コマンドを実行する仕組みを組み込む。 **●工夫・対応の効果** ・運用管理者に対するデータベース管理技術知識の習得や監視を含めた運用管理作業負担が軽減する。 ・想定外のピーク時のオーバフロー予兆監視と対処により、障害の未然防止が可能となる。 ・ディスク容量経年変化傾向を把握し設備計画への反映が可能となる。						○
			18－③ 22－①	③本番稼働システムの異常を早期検知するため、システム内に「一定間隔でのパフォーマンス監視とアラーム(警告)発信機能」を組み込む。資源使用状況により顕著な兆候が見られた際に、当該システム開発部署に連絡を取り、状況を確認する。 **●工夫・対応の効果** 資源不足による障害発生の未然防止に寄与する。						○
			22－①	④本番稼働システムの異常を早期検知するため、システム内に「ログファイルなどの単調増加ファイルの使用率の閾値監視機能」を組み込む。閾値を設定し、外部から定期的に使用率を監視し、閾値を超えた場合に警告表示したり、強制的に予備に切り替えるようにする。 **●工夫・対応の効果** 資源不足による障害発生の未然防止に寄与する。						○
				⑤オンラインアプリケーションのトランザクション性能ログを常時蓄積する仕組みを装備する。 **●工夫・対応の効果** 動作状況監視やトラブル対応、オンライン性能把握に利用可能となる。			○	○		
				⑥一定間隔でのパフォーマンス監視とアラーム(警告)による監視を行う。資源使用状況により、顕著な兆候が見られた際に当該システム開発部署に連絡を取り、状況を確認する。 **●工夫・対応の効果** 資源不足による障害発生の未然防止に寄与する。			○	○		
				⑦JOB監視ツールを導入し、システム使用状況について自動監視を実現する。閾値(複数指定)を超える場合、監視者向けにメールが自動発信される仕組みなども実現する。 **●工夫・対応の効果** システムの異常の早期発見、および資源不足による障害発生の未然防止に寄与する。			○	○		○
				⑧ツールを導入し、システム資源の利用状況(使用容量等)を監視し、月度でレポートを出力する。 **●工夫・対応の効果** システムの利用状況を定期的に監視することにより、システムの異常の早期発見、および資源不足による障害発生の未然防止に寄与する。			○	○		○
			18－③	⑨本番稼働システムの異常を早期検知するため、システム内に「DBの使用容量を監視し、設定した率(70%等)を超えた段階で、アラームメッセージを発生させる仕組み」を組み込む。プログラム内部テーブルについても、同様に使用率を設定し、アラームメッセージを出力させる。 **●工夫・対応の効果** 警告メッセージを使用することにより資源の不足を早期に発見でき、資源不足による障害発生の未然防止に寄与する。			○	○		○
				⑩DBの使用状況のモニタリングを日次出実施する。ベンダ提供のモニタリングユーティリティの処理結果をユーザアプリで加工し、各DBの特性に応じて設定した閾値を元にアラームを表示する仕組みを構築する。 **●工夫・対応の効果** 警告メッセージを使用することにより資源の不足を早期に発見でき、資源不足による障害発生の未然防止に寄与する。						○
			18－③ 22－①	⑪特定業務のトランザクションの一時的な増加による影響を排除するため、あるデータベースの単位あたりの更新回数をモニタリングし、設定した上限閾値以上となった場合、該当するトランザクションを規制する仕組みを構築。一度規制がかかった場合は、更新回数が下限閾値以下になった場合自動的に規制を解除する(この規制の仕組みを手動でも発動可能)。アプリケーションの内部テーブルなどシステム上の制約についても、ドキュメントに一覧化するとともに、各制約値の元となるデータ(テーブルに展開されるレコード件数等)をモニタリングするツールを作成し、定期的に監視・評価する。 **●工夫・対応の効果** 定期的な監視、評価により資源の不足を早期に発見でき、資源不足による障害発生の未然防止に寄与する。						○
				⑫運用部門では毎月パフォーマンスレポートとして、運用状況の監視結果を主管部門に報告する。 **●工夫・対応の効果** ・システムの利用状況を定期的に監視することによりシステムの異常を早期に発見でき、資源不足による障害発生の未然防止に寄与する。 ・資源使用状況を当該システム開発部署に連絡しているため、想定した資源使用状況を開発部署で把握でき、予測と大きく乖離している場合、システムの改善等を実施することができる。						○
				⑬UNIX、LINUX、Windowsサーバについて、ツールにより統一的なリソース監視を行う。予め各リソースについての閾値を設定し、メールの自動発信等の警告を出す仕組みを構築する。 **●工夫・対応の効果** 使用状況に見合った警告メッセージをメール配信することによりシステムの異常を早期に発見でき、資源不足による障害発生の未然防止に寄与する。						○
	障害許容性(拡大防止策) ソフトウェア(又はシステム)の欠陥・故障や予期しない入力があった場合でも、重大な障害や中断を引き起こさず、表面上問題なく動作することができる度合いに関する要求水準(拡大防止)を保全するために講じる事例	停止防止対策		①システムで使用するリソースを分散配置し、障害の影響を局所化する(ボリュームの制約からシステムで使用する端末メッセージキューを同一ボリュームに配置していたが、単一ディスクの障害でオンライン全体が停止してしまった過去障害事例が存在。当該事例をもとに対策を実施)。支店群ごとにボリュームを割り当て、業務ファイルも端末メッセージも支店群ごとに割り当てられたボリュームに配置する。 **●工夫・対応の効果** ディスク障害の影響を局所化することができる。				○		○
				②共通処理の耐故障性を高めるため、ログ書出しがエラーになってもアプリケーションをアボートさせないルールを定める。 **●工夫・対応の効果** 重要処理に影響を与えないような処理については、問題を拡大させない範囲で継続する方針のもと、本番発生障害の抑止が見込める。			○			○
			20－②	③障害発生時には、館内コールで関係者を集め、広く情報共有して影響の拡大を防ぐ。 障害連絡第一報までの時間、暫定対応決定までの時間を定め、SLAとする。 **●工夫・対応の効果** ・障害状況を関係者が認識することにより、迅速な対応が実施できる。 ・障害発生時のSLAを定めることにより、障害発生時の連絡を迅速に行うことが徹底できる。						○
			24－①	①一定期間、既存システム処理を残しておいて、本番処理の中で既存処理結果と新処理結果を比較し、差異があればアラームする仕組みを実装する。 **●工夫・対応の効果** 本対応によって発見された障害は多数あり、早期対応に役立つ。			○	○		○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	障害許容性(拡大防止策) ソフトウェア(又はシステム)の欠陥・故障や予期しない入力があった場合でも、重大な障害や中断を引き起こさず、表面上問題なく動作することができる度合いに関する要求水準(拡大防止)を保全するために講じる事例	稼働初期に起きる故障対策・分析に関する施策	23－①	②カットオーバー直後の本番での実際の運用形態を想定し、試験環境で移行～カットオーバー後数日間の本番リハーサルテストを実施する。 **●工夫・対応の効果** 実際の運用形態に従った確認テストを行うことにより、環境による障害、インタフェースの誤認識による障害の発生を未然に防止する。					○	
			24－①	③既に運用に供しているシステムの機能追加・改修では、不具合発生時に直ちに直前のバージョンに戻せるツールを用意する。(コンティンジェンシープランにのっとった対応を行う。) **●工夫・対応の効果** 不具合発生時早期な対応を行うことができるため、復旧時間の短縮と、障害の拡大の未然防止に寄与する。						○
				④全国の拠点にサーバを配置して稼働するシステムや、多くの機器が集まったシステムでは、ハードウェアの初期不良による影響が大きくなる可能性がある。当該リスクに対する対策として、システムの稼働で特に重要となるサーバ類を早期に購入し、現場の運用形態を想定しながらのヒートラン試験を実施する。さらには、OS、ミドルウエアなど第三者ソフトの開発ベンダのサポートの確保も重要なため、これに対するアプローチも実施する。 **●工夫・対応の効果** ・ハードウェアの部品不良や、導入したサーバのハードウェア部分、OSの不具合も検知することができるため、稼働初期に起きる障害発生を未然に防止することができる。 ・故障ではないが、ヒートラン試験にて発生した各種事象を、障害対応マニュアルに反映することができ、稼働後の保守効率化に貢献できる。						○
		取り扱い可能なデータ件数等システムのキャパシティオーバー時の誤動作防止に関する工夫	25－①	①プロセス間通信用のキューの処理に対し、取り扱うデータサイズによって利用するキューの種類を動的に変更する仕組みを構築する(極力無駄の無いサイズのキューを使い、最適なサイズのキューに空きが無い場合は別のサイズのキューを使う)。これ以外として、「入力規制」、「十分なキャッシュの確保」、「ログ領域のサイクリックな利用」などの工夫も行う。 **●工夫・対応の効果** ひとつのキューが使いきられていた場合にも別のサイズのキューにて処理を継続することができ、障害を発生することなく高負荷状態を乗り切ることができる。						○
				②夜間バッチジョブの障害時、対応方法をあらかじめ登録する。 **●工夫・対応の効果** 急激な件数増加によるスペース不足は、オペレータが対応できるため、発生時に迅速な対処ができる。						○
				③オンラインシステムのピーク時対応に関して、下記対応を行う。 ・アプリケーション処理の適切な多重度設定(スレッド数/プロセス数) ・多重度オーバ時の待ち行列(処理待ちリクエスト)の適切な件数設定(サーバ実装メモリサイズを考慮して設定した件数を超えるリクエストを破棄) ・破棄したリクエストに対し使用者に再入力を依頼 **●工夫・対応の効果** データ件数増加による性能低下を未然に防ぐことができる。			○			
			25－① 26－①	④ホストから拠点サーバに情報を送信する際に、下記対応を行うことにより、取り扱い可能なデータ件数に配慮してシステムの誤動作を防止する。 ・各アプリケーションプログラムで作成した送信データに対して、送信先拠点サーバ毎にデータを集約する。 ・拠点サーバ側に1回の受信可能データ量の制約がある場合、分割送信する仕組みを構築する。送信タイミングを制御することのできるデータはタイミングを遅らせる(翌日に繰り延べる)仕組みも構築する。 **●工夫・対応の効果** キャパシティオーバーによる障害発生を未然に防ぐことができる。						○
				⑤システムで取り扱い可能なトラヒックを超えた場合のエラー処理が重要と判断し、同時実行接続数の上限値を設定し、これを超えるアクセスがあった場合は、利用者にSorry画面を返す仕組みを構築する。実際には新ビジネス用アプリケーションといったトラフィックの予測が困難なシステムに対し適用する。 ・具体例: ピーク時間を昼間帯と見積り、新規Webアプリケーションをリリースしたが、夜間にサーバーがスローダウンし、利用できない状況が長時間続く。実態として、電話夜間割引の時間帯にアクセスが集中していることが判明。 **●工夫・対応の効果** ・トラフィック量の増大による障害発生を未然に防ぐことができる。 ・利用者への配慮が最大の狙いであるが、負荷分散の効果も見込める。				○		
		既存システム修正時における障害(リグレッションテスト漏れ、本番リリース不備による二次障害など)回避策		①以下の障害に対する再発防止策として、テストツールによる自動試験(単体テスト、結合テストに利用)を実施する。 ・テスト期間中に発見した不具合の修正ミスによって引き起こす二次障害。 ・テスト環境から、本番環境に切り替える際の環境設定漏れ。 ・テスト環境でのIPアドレス重複設定など、環境設定に関する人的なミス。 **●工夫・対応の効果** テスト品質の均一化を図ることができる。テスト工数の削減にも寄与する。					○	
				②開発環境にて本番障害対応を行う際に、改良中のシステム資源(プログラム、設計書等)を誤って取り込むことにより、二次障害を引き起こすケースがある。この課題に対し、以下の対応を実施する。 ・改良中のシステム資源は、他システムの開発に影響を与えないよう、別途管理する(構成管理を充実する)。 ・本番リリースを保証する確認テストの実施範囲については、過去のテスト結果を参考にするとともに、上位設計者に確認を得たうえで確定する。 **●工夫・対応の効果** 二次障害の発生を未然に防止することができる。					○	
				③既存システムの修正にて、既存部品(サブルーチン)使用可否の判断誤りで障害を引き起こすケースが多い(異例処理などの妥当性を評価しきれずに使用してしまっている)。本課題に対して、以下の対応を行う。 ・既存部品(サブルーチン)仕様を十分理解したうえで改修を行い、疑問点についてはテストにて確認する。 ・チェックリストに当該事象を盛り込み、再発を防止する。 **●工夫・対応の効果** 既存部品使用に伴う障害は減少し、障害発生の防止に寄与する。			○	○		
				④既存システムの修正では、影響箇所の特定ミスにより、修正漏れが発生し障害を引き起こすケースがある。この課題に対し、以下の対応を行う。 ・影響範囲検索対象を作業者の判断で選別するのではなく、管理対象の全システム資産を対象とする。 ・影響範囲の検索処理は、品質保証部署から全プロジェクトに提供(共通化)する。 **●工夫・対応の効果** 影響調査漏れによる障害発生を未然に防止することができる。			○			

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	障害許容性(拡大防止策) ソフトウェア(又はシステム)の欠陥・故障や予期しない入力があった場合でも、重大な障害や中断を引き起こさず、表面上問題なく動作することができる度合いに関する要求水準(拡大防止)を保全するために講じる事例	既存システム修正時における障害(リグレッションテスト漏れ、本番リリース不備による二次障害など)回避策		⑤個々の修正の影響は微小でも、それが積分されると大きな影響をシステムに与えることがある(アプリケーションの修正版をリリースし事前確認も問題なく完了したが、本番オンライン開始後、バッファ不足でアプリケーションの異常終了が多発するケースなどが起こっている)。本課題に対し、以下の対応を行う。 ・アプリケーションのリリース時にメモリ評価を実施するようリリースレビューポイントを追加し運用する。 **●工夫・対応の効果** 本番リリース前に評価を実施することで、障害発生を未然に防止することができる。				○	○	
				⑥本番障害の対応を行う際、誤って古いプログラムを本番環境にリリースしてしまうことで、本番環境での障害を発生させてしまうケースがある。本課題に対しては、以下の対応(工夫)を行う。 ・オンラインサービスインでは、モジュールを凍結したプレ本番環境で事前確認を行う。 ・本番環境にリリースするプログラムは、リリース前に更新日付・サイズを確認する。 ・サービスイン当日の早朝に打鍵確認を行う。 **●工夫・対応の効果** 障害の拡大(二次障害発生)を未然に防止することができる。						○
				⑦既存システムの修正で新旧比較を行う際、既存システムの不具合により新旧比較で不一致となるケースがあるが、この際に不一致結果の妥当性の判断を誤ってしまうケースがある(修正内容の誤りでありながら、既存システムの誤りと判断してしまう)。この課題に対し、以下の対応を行う。 ・新旧比較結果の評価について、上位設計者(責任者)の判断、合意を得ることを標準化する(チェックリスト化し運用する)。 **●工夫・対応の効果** 作業者レベルの誤った判断がなくなり、障害の発生を未然に防止することができる。				○	○	
			21－④	⑧既存システム改修時での故障抑制として、以下手段を講じる。 ・システム資源の構成管理充実(開発中資産の誤った取り込みを抑止する)。 ・環境設定項目の列挙と、本番環境とテスト環境での差異確認(環境設定漏れ、設定不正を抑止する)。 ・デグレード確認の実施および確認テスト環境準備(必須化)。 **●工夫・対応の効果** 単純ミスによる障害発生の抑止、手戻りコスト低減が図れる。				○	○	
		ハードウェアのアラーム対応に関する工夫	18－④	①各種運用管理ソフトを利用してサーバ・端末に加え、ネットワーク機器についても監視を行う。障害を検知した際にはバトランプを鳴らす。 **●工夫・対応の効果** 運用者が即時に障害を知ることができる。						○
				②運用によって、下記点検作業や状況監視を実践する。 ・毎日の設備巡視による異音・アラーム等の異常確認点検、作業手順の相互確認。 ・随時サーバ状況の監視と定期的情報収集・アラーム(警報)発行。 **●工夫・対応の効果** 運用者が即時に障害を知ることができる。						○
		障害対応マニュアルに関する工夫		①本番実施前にプロジェクト関係者が参加し、コンティンジェンシープランを作成する。 **●工夫・対応の効果** コンティンジェンシープラン作成に、すべてのプロジェクト利害関係者が参加することで、障害発生時には短時間での復旧が可能となる。						○
				②担当者不在のため障害解析に必要なログが取得できないといった事象を回避するため、障害対応マニュアルの記載事項として、各障害箇所に応じたログファイル一覧を記載する。 **●工夫・対応の効果** 復旧作業時に運用者が適切なログを待機することができ、原因究明作業に効果がある。						○
		異常を検知し他システムなどへの影響を遮断する機能に関する工夫		①入力異常を検出した際に、処理結果を出さず、プログラムを停止・再起動させる。 **●工夫・対応の効果** 障害の拡大を防止し、影響を最小限に留めることができる。			○	○		
			28－①	②ネットワーク、サーバ、プロセスなどのシステムの構成要素単位に分けて、単位毎に障害発生、影響範囲、回復方法を設ける。 **●工夫・対応の効果** 障害を局所化することにより、障害の拡大を防止し、影響を最小限に留めることができる。			○	○		
				③セキュリティなどで繰り返し攻撃を受けた場合、あるいは類似したアクセスが生じた場合に、当該トランザクションの処理優先度を下げるか、あるいは分離する。 **●工夫・対応の効果** 障害発生箇所を切り離すことにより、障害の発生を局所化でき、影響を最小限に留めることができる。			○	○		
			27－① 28－①	④プログラム間、プログラム-画面間、プログラム-DB間等のインタフェースとしてやり取りされるデータについて、含まれる項目ごとの型チェック機能等を装備し、誤ったデータが伝搬することを防ぐ。 **●工夫・対応の効果** 無チェックの(ダーティな)データが他システムに拡散することを防止できる。			○	○		
			28－①	⑤特定の処理の異常終了が発生した場合、自動的に規制がかかる仕組みを構築する。 **●工夫・対応の効果** 異常終了の発生時に、事前に予想した影響範囲に規制をかけることにより、影響を最小限に留めることができる。						○
		予防訓練に関する施策		①本番機とは別に訓練用の環境を準備し、異常時の運用訓練を定期的に実施するよう、客先の理解を得る。運用訓練は、実践的な訓練メニューを策定する。 **●工夫・対応の効果** 運用訓練の定着により、異常時の適切な対応が期待できる。						○
			20－③	②定期的に災害復旧訓練の実施。災害シナリオを作成しバックアップ機で対象業務システムの復旧作業と情報伝達の訓練を実施する。復旧処理迅速化のため作業手順や復旧コマンドの簡素化を図る。また、要員育成も兼ねて悪意のある攻撃を想定した演習なども行う。 **●工夫・対応の効果** 災害訓練の実施により、災害の復旧処理の迅速化が見込まれる。						○
				※その他の障害発生を想定した訓練事例 ・法定電源断による停電時訓練。 ・システムテスト工程における障害対応マニュアル(障害時の運用手順)に準じた訓練。 ・外部接続におけるコンティンジェンシープランに沿った障害発生時の対応(接続試験項目に盛り込む)。 ・ユーザ企画による災害やシステム障害を想定した訓練。 ・毎年定期的に有事テストを実施。有事テストの対象は毎年の開発内容を反映し、テスト体制はシステムを設置しているサブセンタ要員と、メインセンタ要員、開発要員が参加。 ・システム開発部門及び運用部門による、定期的なシステム重大障害の発生予行演習。					○	○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	回復性 障害が発生した場合に、性能を回復し影響するデータを復旧できる能力と、それに必要な労力に関する要求水準を保全するために講じる事例	可用性(システムの稼働率、サービス提供時間、運転時間等の割合)向上施策	29－① 30－①	①過去障害(DBミドルウェアプロセスの異常終了時に原因が判明せず)を教訓に、ミドルウェアを監視する仕組みを構築し、当該プロセスが停止していた場合には直ちに再起動する。 **●工夫・対応の効果** 異常終了が発生しても自動的に30秒程度の間に復旧することが可能となり、可用性が向上する。			○			
			29－① 30－①	②アプリケーションの基盤機能として、初期起動か再起動かを判断できる仕組みを構築する。再起動の場合、バックアップファイルから共有メモリ上のデータを復元してからアプリケーションを起動する仕組みを持ち、システム再起動後、オペレーション無しで業務を再開できる仕組みとする。同様にアプリケーションの基盤機能として、アプリケーションの動作状態を監視する仕組みを構築する。例えば、アプリケーションが特定のファイルを一定時間ロックし続けた場合、監視機能が当該アプリケーションを強制終了し、かつ、その後、自動的に再起動する等の仕組みとする。 **●工夫・対応の効果** 再起動に要する時間の削減となり、障害復旧時間が削減できる。			○			
				③メール、ファイル共有システムにおいて、利用者の要請する稼働率に見合うシステム構成(価格)を提示し、バランスをとった(稼働率を下げた)。 **●工夫・対応の効果** ユーザとシステム構成、性能について事前に合意していたため、運用後に問題が発生することがなかった。			○			
			29－①	④DB更新処理では一般に複数の利用者間(プロセス間)でデッドロック状態が発生する可能性があるが、アプリ基盤で自動的にトランザクションのロールバックとリトライの処理を実行する機能を提供する。 **●工夫・対応の効果** 各アプリに負担をかけずに、稼働率向上を図ることができる。			○			
				⑤バッチ稼働時間の削減として、以下の処置を実施する。 ・SORTの工夫。 ・JOBの分割化。 ・データアクセスの変更。 ・バッチJOBの分割と実行多重度のチューニング。		○	○			
				⑥稼働率の向上を目的に、以下の対応を行う。 ・DBバックアップ処理のコンカレント化。 ・アプリケーションホストの2系統化および端末－ホスト間の2系統化(1系統ダウン時も業務継続可能)。 ・バックアップセンタの設置。		○	○			
		障害分類を配慮した障害回復時間短縮施策		①バッチ処理が異常終了した時に、単純にリランすれば、異常終了した時のデータをスキップして処理する仕組みを作成する。スキップしたデータはログに出力する。また、スキップの論理も、異常終了したデータだけスキップする、異常終了した処理コード全てを対象にする、異常終了した証券番号の全ての処理を対象する等、指定可能とする。 **●工夫・対応の効果** スキップしたデータはログに出力しているので、処理終了後(翌日)に障害解析でき、夜間呼び出しが回避されると共に、障害復旧業務効率が向上した。			○	○		
		サービス提供再開施策(復旧予測時間の配慮など)	11－③ 33－②	①障害発生時の対応については、業務回復優先とした体制、手順を準備する。 **●工夫・対応の効果** 障害解析チームと業務回復チームに分けて対応することにより、重複した作業を行わず効率的な回復を行うことができる。						○
				②オンラインシステムの機能ごとにサービスの停止/再開を指示(コントロール)できる機能をアプリ基盤として提供する。 **●工夫・対応の効果** 障害を局所化することにより、他システムへの拡大を未然に防止する。			○	○		
			21－①	③発生した障害により、以降どのような重要処理・イベントが影響を受けるか、検索できる仕組みを構築する。 **●工夫・対応の効果** 復旧作業のタイムリミット、優先度が的確に判断できる。						○
		自動リカバリ機能など故障耐久能力向上施策		①取引毎にリアルタイムに更新する仕組みと、1日の終了時の静的DBから再構築する仕組みの二本立てとし、リアル更新に障害があっても、1日たてば回復するできるような仕組みを構築する。 **●工夫・対応の効果** リアル更新に障害があっても、1日たてば回復できるため、業務への影響(与信枠を超えた受注や出荷)を抑えることができる。また、両者の処理の差異はログが出力されるので、それをきっかけに障害を検知でき、障害解析の効率化にも寄与する。			○	○		
				②ホストオンラインシステムにおいて、トランザクションがデッドロックにより異常終了した場合は、自動的に再スケジュールしてリトライを行う。 **●工夫・対応の効果** デッドロックによるレスポンスの低下、障害の拡大を未然に防止することにより、性能要求を実現する。			○	○		
			20－④	③自動リカバリ機能として下記対応(工夫)を行う。 ・ディスクはRAID構成(ミラー構成あるいはホットスペア構成)として、障害発生時の復旧の自動化を行う。 ・ネットワークはルータ、スイッチ、経路等を二重化して障害時は迂回する。 ・処理単位にコミットポイントを設け、データを一部更新した状態で障害が発生した場合、所定の処理単位までリカバリを可能にする。 ・登録・更新系SQLをログ出力しておき、障害発生時は朝時点のDBからSQLを利用して復旧させる。 ・アプリケーションプログラム起因による異常終了(※)の場合、停止したプログラム稼働領域は自動リカバリする(ミドルウェアの機能)。 ・オンライン稼働中にオンラインアプリケーションプログラムの入替が可能(同上)。 ※ アプリケーションプログラム内で論理矛盾等の異常を検知した場合は、ミドルウェアに制御が渡り、当該プログラム稼働領域の異常終了と回復が行われる。なお、異常終了後に自動リカバリさせるか否かはアプリケーションプログラム側で指定することが可能(ここでいうミドルウェアはアプリケーションプログラムとDBマネジメントシステムの間に位置するものでアプリケーションプログラムの制御を支援するもの)。 OSやDBマネジメントシステムが障害検知したことに伴う異常終了は、当該ミドルウェアが介在しないため自動リカバリできない(この場合は、運用自動化ソフトウェアが異常終了を検知し、必要なコ			○	○		○
				④自動リカバリ機能で使用するリソースの配置や障害に留意する。 ・自動リカバリ機能のために資源を確保する(DB回復のためのログファイルなど)。 ・業務ファイルとリカバリのための資源は分散配置する。 ・リカバリのための資源に障害が発生(二重障害)が発生した場合を想定しておく。			○	○		○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
信頼性	回復性 障害が発生した場合に、性能を回復し影響するデータを復旧できる能力と、それに必要な労力に関する要求水準を保全するために講じる事例	バックアップ機への切り替えに関する施策	31－①	①バックアップ機について「定義ファイルの変更(テスト機として利用時の変更)」や「プログラムバージョンの更新漏れ」などにより、切替時にシステムが立ち上がらないといった弊害を招くことがあるため、通常保守および運用において、定義ファイル確認やプログラムバージョン管理を徹底する(プログラムライブラリはディスクミラーリングで同期をとる)。 **●工夫・対応の効果** 本番機切替時の障害を未然に防止する。						○
				②バックアップ機へ切換えた際に、バックアップ機側のプログラムのバージョンが古かったため、正しく動作しなかった。原因は、プログラムの修正が発生した際に、バックアップ機へのバージョンアップをしていなかった。 **●工夫・対応の効果** 本番機のバージョンアップ時にも、バックアップ機へも反映することについてルール化しておくことにより、更新漏れを防止する。						○
				③(運用)メールシステムのバックアップ環境を提供したが、バックアップ環境との切り替え時に既読未読情報が引き継がれない(仕様)ことが事後に判明。当該方式のメリット(自動的に素早くバックアップ機に切り替わること)を優先し、諦めてもらった。 **●工夫・対応の効果** コスト削減のため、機能に優先順位をつけ、実施する(障害回復と可用性のトレードオフ)。						○
			18－⑤	④サーバの冗長化設計をする場合には、縮退運転時の保証機能、性能劣化について、運用ルールを定め、ユーザと合意する。 **●工夫・対応の効果** 縮退運転に関し、ユーザと事前合意を行っておくことで、運用時の障害発生を未然に防止する。						○
		広域・局所災害対策		①災害対策用に、別地区に災害対策用のシステムを構築する。 ・災害対策用システムは非冗長化構成をとり、コスト削減を図っている。 ・定期的に本番機とデータの同期をとり、整合性を保つよう配慮している。 ・遠隔地へのバックアップセンタの構築。 ・ユーザに災害対策を施すよう合意するとともに、災害対策に則った規則を作成する。 ・災害発生時は、その規則に従いシステムの切替を実施する。						○
			30－②	②データベース構成やデータの保管について、以下の対策をとる。 ・大規模データベースを拠点別(支店別)に区画分割構成することによる障害の全体への影響波及や相互干渉の防止。 ・現場業務継続続行に必要な必要最低限のデータを事業所に退避保管。定期的に更新し災害時の情報検索業務が可能な仕組みを整備。 **●工夫・対応の効果** 災害発生時に影響を局所化する。						○
				③遠隔地バックアップセンター設置にあたり、災害時に継続すべき業務内容を絞り込み、バックアップセンターの効率化を図る。 **●工夫・対応の効果** バックアップセンターでの業務を絞り込み事前にユーザの合意を得ておくことで、最低限のシステム構成での対応で、最大限の効果を期待することができる。						○
		要件定義時点におけるユーザとベンダ間のコンティンジェンシープラン共有化		①業務継続性を実現するための対象業務範囲である、耐故障を実行する基本戦略、誤り検出(誤りの存在の特定)、回復／誤り処理(システム状態から誤りを除去する)、回復／故障処理(故障の再発防止)について、お客様と事前に合意する。 **●工夫・対応の効果** 要件定義時にコンティンジェンシープランを作成することにより、システムの信頼性について事前に合意が得られ、テスト実施時のコストが明確になる。		○			○	○
				②要件検討時点以外でのコンティンジェンシープランを作成し、ユーザとベンダ間で合意する。 ・全ての内容を要件定義時に確認することは難しい場合や、テストを実施しないと決定が難しい内容については、詳細を試験によって定めることを合意する。 ・本番実施時にコンティンジェンシープランを作成することが義務付けられており、ユーザの合意も必須とする(要件定義時点では作成しない。実施手順作成時に作成)。 **●工夫・対応の効果** コンティンジェンシープランについて事前合意しておくことにより、実際の発動時速やかに実施できるため、障害復旧時間の短縮となり、影響の拡大を防止することができる。					○	○
			19－②	③コンティンジェンシープランとして、移行失敗時に旧システムに戻す時期や判断基準を設ける。 ・利用者数の多いシステムを対象とする。 ・事前取り決め範囲:旧システムに戻すか対策前進するかを決断する有識者、 判断条件(対策前進時のタイムリミット、旧システムに戻す場合の切替時期やそれに伴う付加費用に付随する問題)。 **●工夫・対応の効果** 実際の発動時速やかに実施できるため、障害復旧時間の短縮となり、影響の拡大を防止することができる。					○	○
	信頼性標準適合性	信頼性に対応する規定(業務、内部統制、ISMS、国際会計など)および開発標準の適合に関する監査対策		①事業継続に関する分野の規格・基準・ガイドライン等 ・BS25999(英国規格協会)、事業継続計画策定ガイドライン(経済産業省)、中小企業BCP策定運用方針(中小企業庁)、金融機関等におけるコンティンジェンシープラン策定のための手引書(FISC)、第2次情報セキュリティ基本計画・重要インフラの情報セキュリティに係る第2次行動計画(安全基準等、IT障害の予防的対策と事後的対策)(内閣官房情報セキュリティセンター、2009年) 信頼性に関する分野の基準・ガイドライン等 ・情報通信ネットワーク安全・信頼性基準(昭和62年郵政省告示第73号)、情報システムの信頼性向上に関するガイドライン第2版(経済産業省、2008年) ・ISO/IEC9126(JIS X0129):ソフトウェア品質特性 ・IEC 60300-1/2(JIS Z8115):ディペンダビリティ ・ISO/IEC15288 (JIS X0170):SLCP(システム) ・ISO/IEC12207(JIS X0160/共通フレーム2007):SLCP(ソフトウェア) ・ISO・PC236/ISO21500:PMBOK ・ISO15504:CMMI ・ISO/IEC 38500:COBIT これらに対する準拠が、調達仕様上に盛り込まれる。 **●工夫・対応の効果** ・法令・規格・基準・ガイドライン等より、遵守しなければならない内容、特に誤解しやすい内容をチェックリストに整理し、運用することで法令遵守に対しコスト削減を図る。 ・法令等より定期的に問題集を作成し、実施することでプロジェクト参加者の法令等の理解状況を確認する。		○				○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性（2次）	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
使用性	理解性 利用者がソフトウェア（又はシステム）を理解するために必要となる労力度合いに対する要求水準を保全するために講じる事例	ユーザインタフェース（メニュー、アイコンなど）の工夫	13－②	①ユーザインタフェースの視点に立ち、下記対策（工夫）を実施する。 ・開発のベースとなるフレームワークにおいて画面制御内容に制限を設けているが、その制限を踏まえた画面設計を当初から行い、業務に影響がないことを確認しつつ設計を進める。 ・イレギュラーな操作／入力に対する適切なメッセージを表示する（利用者がメッセージから誤った入力／操作を把握可能であること）。 ・各業務画面の設計に先立って、共通ユーザインタフェース機能（画面内の共通レイアウト部分のデザイン、利用を許可する画面コンポーネント、項目配置ルール、など）について当該システムに相応しいユーザビリティを実現するための設計規約／設計基準を確立する。 **●工夫・対応の効果** ユーザの利用しやすさが向上し、誤入力による障害発生を未然に防止する。			○			
			13－②	②ユーザインタフェースに関連するドキュメントやツールなど情報発信する体制を確立し、Webシステムのユーザビリティ向上を目指す。 ・ユーザビリティの評価手法として、「ヒューリスティック評価」を採用。 ・さらに、特殊な技術を持つユーザビリティ評価の専門家でなくても簡単に画面を見ながらチェックできるような、“ユーザビリティチェックシート”を策定。 ・ユーザビリティを高めるためのポイントを体系的に整理し、基礎知識さえあれば誰でもがユーザビリティを評価できるようにする。 **●工夫・対応の効果** ユーザの利用し易さが向上し、誤入力による障害発生を未然に防止する。			○			
				③情報システム基盤の実現要求をお客様視点で見える化する共同検討会「発注者ビュー検討会」（http://www.nttdata.co.jp/cview/）の発注者ビューガイドライン（画面編、システムの振る舞い編、データモデル編）に準拠する。 **●工夫・対応の効果** プロジェクト参加者の役割分担、責任範囲が明確となるため、作業の重複を防止する。また、レビュー観点が予め提示されることにより成果物の精度が上がり、スケジュールの遵守が期待できる。		○	○			
		利用者側の教育効果向上施策	6－② 23－②	①利用者の教育のため、操作訓練等のトレーニングを実施する。 ・重要データの入力等、通常はなかなか実践できない業務に絞って入力練習のメニューを構築する。 ・一日の流れにそって全てのノーマル業務を実行できる試験モードを構築する。 ・本番環境に研修用DBを構築。本番機よりデータを抽出し小規模のデータベースを構築し、本番機同等の業務処理のトレーニングができるようにする。 ・システム利用者向け、研修環境の構築と教育を実施する。 ・システムのサービスメニューとして「研修」をシステムのエンドユーザに提供する。（基本的な操作パターンを準備し、本番データにはアクセスせず、ダミーの帳票出力などを行う）。 **●工夫・対応の効果** 教育環境を準備することで、利用者のスキルが向上し、誤入力による障害発生を未然に防止する。		○				○
			6－②	②利用者の運用補助のため、下記対応（工夫）を行う。 ・ヘルプデスクの設置 ・運用手順の省略（自動化製品の利用推進）とマニュアルの充実 ・上述した共通ユーザインタフェース機能を利用者が直感的に理解しやすい一貫性のある規約・基準として確立する。 **●工夫・対応の効果** 利用者のスキルが向上し、誤入力による障害発生を未然に防止する。						○
				③外部との接続試験実施時に実際のユーザに打鍵してもらうことにより、教育を兼ねる。 **●工夫・対応の効果** ・接続試験実施時にユーザ打鍵を行うことにより、事前に操作手順の修正を反映できるとともに、本番運用時の操作ミスの削減に寄与する（教育用手順書はユーザが作成）。 ・実際の画面を使用し、画面遷移を記述した教育用資料を使用し、テスト打鍵することで教育工数を削減する。						○
	習得性 ソフトウエア（又はシステム）を使用する上で参照するマニュアル等の使いやすさ、運用方法、習熟時間等に対する要求水準を保全するために講じる事例	ユーザレベル（初級、中級、上級）などを配慮した習得容易性向上施策		①段階を追って画面遷移する、画面遷移をせずダイレクトに画面IDを指定して画面を表示（ジャンプ機能）する等、使用者の経験度合いにより使用時間の短縮が図れるような仕組みを実装する。提供する画面遷移のパターンは、ユーザのレベルおよび構成比率を考慮して、必要最低限に絞り込んだ。 **●工夫・対応の効果** ユーザの利用し易さが向上し、誤入力による障害発生を未然に防止する。			○	○		
				②実際の画面を使用し、画面遷移を記述した教育用資料を使用し、テスト打鍵することで教育工数を削減する。 **●工夫・対応の効果** 実際の画面のため操作手順がわかりやすく、操作ミスの削減に寄与する。			○	○		
			23－①	③使用性（ユーザビリティ）の向上施策として、標準ルールを制定する。（以下はキーボード操作ルール） ・高年齢の方向けにはキーボード操作のみとする。 ・キーボードは利用するキー以外を無効にする。 ・基幹業務系システムにおいては、最終的にキーボードによる連続入力の操作性が要求されることを考慮し、マウスによる操作以外に、キーボードによる操作に対応する。 ・処理までのキータッチ数に対し、閾値を設定（多い場合はタッチ数を削減するよう変更）する。 **●工夫・対応の効果** 習得し易くなることで、誤入力による障害発生を未然に防止する。			○	○		○
		システム利用マニュアルおよびシステム運用マニュアルを判り易くするための工夫		①単純だが、1ページに1画面を基本（画面イメージを大きく貼り付ける）とし、画面上に吹き出しA、B、C・・・と説明を入れ、操作の流れと操作方法を入れたマニュアルを作成する。本マニュアルを使って、教育（集合研修）を行う際も、丁寧に説明することができる。 **●工夫・対応の効果** マニュアルを読むだけで理解できる形としたうえで、説明も行うことにより、運用後の操作ミスの低減に繋がる。			○	○		
				②見やすく誤りのないマニュアル提供のため、下記対応（工夫）を行う。 ・テクニカルライティングの専門家に依頼する。 ・システム開発関係者以外の第三者にマニュアルのレビューを依頼する。 ・利用者からの問合せを分析し、マニュアルの改善に繋げる。 ・メッセージの内容、対処方法などをマニュアルに記載する。 **●工夫・対応の効果** 運用後の操作ミス削減、運用担当者の問い合わせ負荷低減が図れる。			○	○		○
				③各業務画面にヘルプボタンを配置して業務画面の使い方がわかるよう工夫する（マニュアルレスの運用）。 **●工夫・対応の効果** マニュアルを読まずとも運用に入れるため、理解のための工数削減に寄与する。			○	○		○
		ヘルプ機能の容易性などに関する工夫	13－③	①画面表示について、以下の工夫を行う。 ・特に監視画面にて、参照したいデータ表示エリアにマウスを移動するとその項目名を表示する。 ・データで表示されているデータ表示エリアにマウスを移動すると、そのデータの翻訳名を表示する（例えば、6503が「●●会社」をあらわすようなデータがあったとして、6503の上にマウスを移動すると「●●会社」と翻訳されて表示される等）。 **●工夫・対応の効果** 画面表示エリアを効果的に使用しているため、表示データが多量なときに有効であり、説明文もあわせて表示されるため、コードのみのときに比べ、操作ミスが低減する。			○	○		
				②ある言語ソフトのヘルプの中に実際に稼働するコードをセットし、それをそのまま流用できるようにする。ヘルプは、該当機能の内容だけでなく、場合によっては、他の機能との関連の説明もつける。 **●工夫・対応の効果** 関連する説明が表示されるため、操作漏れの減少が期待できる。			○	○		
				③PCのプルダウン機能と同様に、ホスト機で、ファンクションキー押下によるHELP画面の表示を実現する。 **●工夫・対応の効果** プルダウンでの入力を可能としているため、入力ミスが低減する。			○	○		

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
使用性	運用性 ソフトウェア(又はシステム)の操作に関わる操作効率、操作状況認識などに対する要求水準を保全するために講じる事例	利用者に対する操作のナビゲート、操作結果の確認のし易さに関する工夫	6－③ 13－④	①メニュー体系について、そのとき使えるメニューだけをアクティブ状態にする工夫を行う。 A. 本番・試験選択→B. 業務開始→C. 情報確認→D. 入力→E. 業務終了という流れが必要であった場合、A. を実施するとB. がアクティブになり、B. を実施するとCがアクティブになり. . . という仕組みを構築する。 **●工夫・対応の効果** 利用手順が明確になるため、操作ミスが低減する。			○	○		
				②システム毎の操作性差異やばらつき等を抑制するため、画面デザインガイドを規定、操作性の標準化(色使い、文字、子ウィンドウの使用方法等)を図る。 **●工夫・対応の効果** 業務や部門間でのデザイン差異を抑制することにより操作性が向上する。			○	○		
				③データエントリーにおいて、メニューに戻らずに次々にエントリー画面に遷移する仕組みを構築する。画面遷移機能を共通化し、アプリケーションでは意識しないようにする。 **●工夫・対応の効果** 大量のデータを入力する場合の操作性が向上する。			○	○		
				④入力画面と確認画面のレイアウトを一致させて操作者が入力結果を視認しやすい画面設計基準を設ける。インターネットサイトでは全く別のレイアウトとしている事例も多いが、確認画面の各項目は入力画面の各項目をプロテクト状態(入力不可状態)に状態変更するようにし、画面レイアウトは共通(同一)のものを使用する。 **●工夫・対応の効果** 利用箇所が明確になるため、操作ミスが低減する。			○	○		
			7－④	⑤お客様が使用する機能の改定時には、オーナーだけでなく、ヘルプデスク・コールセンターにも要件を確認し、意見を取り入れる。 **●工夫・対応の効果** 実際の使用者からの意見を取り入れることに繋がるため、運用後の改善要求の削減を図れる。		○				
		システム運用の容易性向上、誤操作防止施策		①日常のシステム運用作業は、制御用端末を設けて、すべて、この端末から実施できるように統合管理をする。 **●工夫・対応の効果** すべての作業が制御用端末より行えるため、作業効率が向上する。			○	○		
				②運用において、以下対応を行う。 ・運用手順の省略(自動化製品の利用、機能の作り込み)を図る。 ・マニュアルを充実する。 ・ライブラリ管理者の資産受付、リリースなどの操作においては、操作後の結果ログ(扱った資産名、処理内容など)がポップアップ画面で即時確認を可能とする。 **●工夫・対応の効果** パッケージ製品を効率的に使用することにより、短期間での構築が期待でき、効率的な運用に寄与する。			○	○		
			28－②	③運用のスタンダードを記載した「運用設計チェックシート」を開発段階で作成し、運用部門のレビューを受ける。 **●工夫・対応の効果** 標準外の運用を極力削減し、運用部門と開発段階で調整を行うことで運用トラブルの削減を図る。			○	○		
				④データのバックアップ方式の統一化を図ることは可能であるが、実際のバックアップや別地保管の詳細な方式をマニュアル等で開発プロジェクトや運用部門の各メンバーに浸透させることは非常に困難なことである。 バックアップでの人的作業ミスを防止するため、バックアップやリストア手順の統一化を図るとともに作業手順をスクリプト化し提供する。 **●工夫・対応の効果** 運用作業が標準されることで、運用トラブル(人的作業ミス)の削減を図る。			○	○		
		インストールやバージョンアップの容易性(バージョンアップ後の確認の容易性、配布容易性を含む)向上施策	32－①	①ソフトウェアの自動配信機能を構築する。拠点サーバのプログラム入れ替えは、通常は自動配信機能を使用し、大量のプログラム入れ替えではCD－R等のメディアを現場へ郵送して行う。 どちらの手段でも、新機能リリースの1週間前から更新を開始し、済／未了の監視をメインフレーム側で監視できる仕組みを構築し、2日前には未了の現場へ督促する態勢にある。 なお、サーバのアプリプログラムは、リリース日により処理ロジックが制御される仕組みであり、リリース日の前に新機能が動くことは無い(テストで事前に確認済み)。 **●工夫・対応の効果** 配信機能を自動化することにより、人手が介在することに起因するミスを未然に防止できる。						○
				②バージョンアップによる障害を防止するため、下記対応(工夫)を行う。 ・原則システム更改以外のバージョンアップは行わない。 ・構成管理、リリース管理を充実する(運用中システムのバージョンレベルも管理する)。 **●工夫・対応の効果** ソフトウェアのバージョンアップ(誤作業)に伴う障害発生の未然防止に役立つ。						○
				③セキュリティパッチやクライアント側への配布ソフトウェアは自動配信し、ユーザの操作を不要とする。 **●工夫・対応の効果** 配信機能を自動化することにより、人手が介在することに起因する漏れ、ミスを未然に防止できる。						○
		誤り修正(取り消し作業を含む)の容易性向上施策		①誤操作を取り消す場合に直近の操作から順番に取り消さねばならない制約を持つシステムについて、対象の取消操作の頻度や操作効率化による削減コストを評価し、顧客と合意のうえ、制約付で一気に戻す取消機能を作った。 **●工夫・対応の効果** 過去まで遡る作業が不要となり、誤り修正が容易となる。			○	○		
		誤操作からの回復性向上施策	7－①	①削除要求時実際に削除せず、削除の取り消し機能を提供する。 **●工夫・対応の効果** 入力データのトレーサビリティが明らかになるとともに、誤操作による障害の発生を防止する。			○	○		
			7－①	②エラー画面の標準化について、以下の画面設計基準を設ける。 ・複数個のエラー項目をすべて赤表示⇒エラー項目の視認性。 ・赤表示されたエラー項目を操作(カーソル位置付け)するタイミングでエラー説明メッセージを動的表示⇒修正入力が行いやすい。 ・利用者の理解しやすい表記とする(システム管理者的表現は避ける)。 **●工夫・対応の効果** 入力ミスが明確になるため、操作性が向上する。			○	○		
		利用時のメッセージ理解性向上施策		①「エラーコード」、「発生事由」、「発生原因」、「対処方法」を必ず定義する。 **●工夫・対応の効果** マニュアルに上記内容を明記することにより問い合わせが減り、運用部門の負荷低減に寄与する。			○	○		
			7－②	②メッセージ出力について、以下の工夫を行う。 ・赤表示されたエラー項目を操作(カーソル位置付け)するタイミングでエラー説明メッセージを動的表示⇒修正入力が行いやすい。 ・利用者の理解しやすい表記とする(システム管理者的表現は避ける)。 **●工夫・対応の効果** エラー箇所に理由が明記されるため問い合わせが減り、運用部門の負荷低減に寄与する。						○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
使用性	運用性 ソフトウェア(又はシステム)の操作に関わる操作効率、操作状況認識などに対する要求水準を保全するために講じる事例	オペレータの介入操作に関する工夫		①手順、申請などのルール明確化。 **●工夫・対応の効果** 手順、ルールを統一することで問い合わせが減り、運用部門の負荷低減に寄与する。						○
			28－③	②運用部門担当者が操作を実施またはシミュレーションし、操作のし易さ、妥当性を検証する。 **●工夫・対応の効果** 運用部門の負荷低減に寄与している。						○
	魅力性	少ないオペレーションで多くの処理を実現する工夫		①当日業務にて発生した申し込みデータについて、一覧形式で一括変更・削除・承認する機能を提供する。 **●工夫・対応の効果** 申し込みデータ毎でのオペレーションが不要となり、利用者の利便性も向上する。なお、		○	○			
	使用性標準適合性	使用性に対応する規定(業務、内部統制、ISMS、国際会計など)および開発標準の適合に関する監査対策		①ISO/IEC9126(JIS X0129):ソフトウェア品質特性。 **●工夫・対応の効果** 規格より、ソフトウェアの使用性に影響を与える項目をチェックリストに整理し、運用することでコスト削減を図る。		○				○
効率性	時間効率性 処理時間(レスポンスタイム、スループット、ターンアラウンドタイム)および処理能力(CPU、主記憶、ファイル、回線の使用率等)の要求水準を保全するために講じる事例	非平常時(ピーク時、障害および災害時など)に対応するレスポンスタイム、スループット、ターンアウンドタイムの考慮点などに関する施策		①使用頻度が高いDBは、メモリー常駐する仕組みを採用する(製品を導入したわけではなく、自前でメモリー常駐する仕組みを作る)。DBがDISK上か、メモリー上かをアプリケーションが意識しなくても良いよう、DBインタフェースを共通化する。(DISK→メモリーもプログラム修正を必要としない)。 **●工夫・対応の効果** ピーク時での性能低下(レスポンスタイム悪化)の事前対策にもなり、性能維持に寄与する。			○	○		
				②ロードバランサーを経由して接続するシステムでは、障害時は、障害が発生したサーバを切り離して縮退運転を行う。縮退運転においては、レスポンスタイムなどに影響があることを規定する。自動ローディング機能を利用して負荷を与え、処理時間を計測・集計を行う。 **●工夫・対応の効果** 本番稼働前に、ユーザと縮退運転時の品質、ピーク時の処理時間予測に関し合意を得ることができるため、性能面での不安が解消される。			○	○		
			25－②	③業務アプリ設計者自身が設計中の機能についてピーク時も考慮したオンライン応答時間／バッチ処理時間を容易に見積れるツールを提供した。提供ツールは、資源使用率等も把握するため資源のサイジング情報も同時にアウトプットする形式とした。 **●工夫・対応の効果** ・設計の早い段階で性能要求を満たせるか否かを把握できるため、顧客との共通認識のもと手戻りリスクも減らすことができる。 ・負荷分散構造システムによる異常時の考慮を事前に行うことにより、性能低下を未然に予測することができる。			○	○		
				④バッチ運行のJOBネットやオンライン業務システムのセットアッププラン等の設計時、システム規模の大きさや複雑度が増すにつれてジョブ間のクリティカルパス等を見失いがちになること配慮し、クリティカルパス及び並行処理可能なプロセスの観点で横断的にチェックを実施する。 **●工夫・対応の効果** 処理時間の短縮が図れる。			○	○		
				⑤トランザクション件数のピーク＝システム負荷のピークとしてとらえず、トランザクションの処理の重さ(ダイナミックステップなど)を考慮し、設計を行う。 **●工夫・対応の効果** 当該工夫(設計)により、ピーク時評価の精度を向上させることで、性能低下を未然に防止できる。			○	○		
		非平常時(ピーク時、障害および災害時など)に対応する業務のスループット、デリバリータイムの考慮点などに関する施策		①ピークも様々なので、前提条件を定めて、性能要件を規定する。 **●工夫・対応の効果** 事前に様々な条件を規定することにより、ユーザとの性能要件に対する合意が容易となる。		○	○			
				②業務運用で回避できるピーク(バックオフィス業務)と回避できないピーク(ATMやフロント業務)を区別しておく。 **●工夫・対応の効果** 事前に様々な性能低下条件を規定することにより、ユーザとの性能要件に対する合意が得やすくなるとともに、対応までに要する時間の短縮が期待できる。		○	○			
	資源効率性 ソフトウェア(又はシステム)機能を実行するのに使用する資源の程度(資源使用量、資源使用率など)に対する要求水準を保全するために講じる事例	非平常時(ピーク時、障害および災害時など)に対応する資源(CPU、メモリー、伝送、入出力、設置条件(スペース、電源容量など))、体制などの考慮点に関する施策	26－②	①仮想化技術やパーティショニング技術によって、負荷量が大きく変動した時に動的にCPUリソース等を増強・配分変更できる仕組みを構築する。また、スタンバイ機等も組み込むなど有効活用の仕組みを構築する。 **●工夫・対応の効果** 高性能のシステム構成を最小限のコストにて実現できる。			○			
			26－②	②DBを可変長レコード定義し、DBインタフェースで圧縮展開機能を組み込んでDB格納量を削減する。 **●工夫・対応の効果** ディスク使用量が減少するとともに、性能も向上する。			○			
				③系切替時にはいっせいにログインが集中し、リソース不足となりやすいので注意が必要である。(系切替したがログオン集中のためリソース不足で再度ダウン、これを繰り返した経験あり。)			○			
	効率性標準適合性	効率性に対応する規定(業務、内部統制、ISMS、国際会計など)および開発標準の適合に関する監査対策		①SLAに関する分野のガイドライン等。 ・公共ITにおけるアウトソーシングに関するガイドライン(総務省、2004年)、民間向けITシステムのSLAガイドライン(第3版)(日本情報技術産業協会(JEITA))、情報システムに係る政府調達へのSLA導入ガイドライン(経済産業省)、ASP・SaaSにおける情報セキュリティ対策ガイドライン(総務省、2008年)。 ・データセンターの安全・信頼性に係る情報開示指針＜PUE(Power Usage Effect:グリーンITの共通指標)等環境対応指針含む＞(総務省、2009年) ・ISO/IEC9126(JIS X0129):ソフトウェア品質特性。 **●工夫・対応の効果** ガイドライン・規格より、ソフトウェアの効率性に影響を与える項目をチェックリストに整理し、運用することでコスト削減を図る。	○					○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
保守性	解析性 障害原因の識別、欠陥特定、欠陥対処などの効率化に関する要求水準を保全するために講じる事例	データログ実装、状況監視データ取得など活動記録保有能力に関する施策		①データログ実装時、下記対応(工夫)を行う。 ・端末の操作履歴を残す仕組み構築。 ・ログの二重保管。 ・設計時に、ログの増加量の推定と保存期間の設定を実施するとともに、システム試験において実環境に近い形で検証を行う。推定が間違っていた場合には、方式変更(圧縮や別媒体への退避等)を行う。 **●工夫・対応の効果** ログ収集が容易となり、有事での解析効率が向上する。			○	○		
			33－②	②以下のようにログ方式の型決め(標準化)を行う。 ・ログデータの目的の整理と分類の方法。 ・各アプリ実装時のログデータ取得方法。 ・システム運用時のログデータ参照方法。 ・業務アプリケーション実行時のログ収集(アプリログ)。 **●工夫・対応の効果** ログ収集の標準化を図ることにより、システム開発時に新たな仕組みを検討する必要がなく、開発効率の向上が図れる。			○	○		
				③インシデントの分析(そもそも何が原因で問題が起こったのか)を行い、表面的な対応をするのでなく、根本原因に対する対策を講じる。 **●工夫・対応の効果** 効果的な対応策を図ることができ製品品質の向上に寄与する。			○	○		
				④ログ出力先が一杯になったら、別のところにスイッチし古いログをバックアップに移す仕組みを構築する。 **●工夫・対応の効果** 過去のログを失うことなく保存しているため、障害発生時のログ解析に有効に利用できる。			○	○		
		診断機能実装および故障原因解析に関する工夫		①処理時間が閾値を超過時にログに出力する機能などを埋め込む。 **●工夫・対応の効果** 故障原因、性能低下を把握することができ、障害の拡大を未然に防止することができる。			○	○		
				②アプリケーションのエラー処理基準を設定する(アプリケーションではエラー検出時に一律異常終了するなど軽視されがちであること、エラー解析をOS等のコアダンプから解析するのは困難であることに配慮する)。アプリケーションでエラー検出した場合は、エラー箇所を特定可能なデータをスナップ取得するなど、エラー解析が容易となる仕組みを構築する。 **●工夫・対応の効果** 有事での障害解析効率が向上する。			○	○		
				③システムサーバ稼働状態情報の定期的(数分間隔)収集と異常時のアラーム警報(例として、pingを打って、レスポンスがない時にアラームを発する等)を行う。 **●工夫・対応の効果** 障害発生、性能低下を早期に把握、特定することができ、障害の拡大を未然に防止することができる。			○	○		
		ソフトウェアのトレース機能など解析容易性向上施策		①アプリケーション内で異常終了する際のメッセージに場所を特定する項目を入れる(プログラム名＋プログラム内SEQ)。 **●工夫・対応の効果** 想定したステップ数より増加したが、障害原因が特定できるため、解析時間の短縮が図れる。			○	○		
				②バッチ処理において、データレコード生成、更新時に、更新元機能を識別するIDを埋め込み、データを手がかりに障害を引き起こしたプログラムの特定を容易にする。開発時にも、トレース機能(どの段階で、サブルーチンを呼んだか、DBからどのようなデータをアクセスしたか等々の情報を収集している)によるテストを可能とすることで投資コストは回収できる。 **●工夫・対応の効果** 受注、出荷、在庫など様々なサブシステムで、同一のファイルにデータを生成する場合、障害原因元を特定するのに有効である。			○	○		
			17－③ 29－②	③アプリケーション基盤機能にて、プロセス間のキュー通信情報を全てログに残す仕組みを構築する。 ・アプリケーション側は単純にキュー送信関数をコールするのみで、基盤機能が自動的にログを残す仕組みとする(その際、送信元プロセス名、宛先プロセス名等も記録することで解析が容易になる)。 ・アプリケーションがトレースとは別に実行履歴を残す、いわゆる標準出力ログについて、アプリケーションのファイル名(_FILE_)・行番号(_LINE_)を出力することとする。 ・プログラミング後に自動的に、ログ出力を挿入する仕組みを加える。 ・入力トランザクション毎に一意の番号づけ、発生時刻を刻印して、トランザクションがシステムから消滅するまでデータ中に保持する。 **●工夫・対応の効果** 的確なログを採取することにより、障害原因が特定できるため、解析時間の短縮が図れている			○	○		
			29－②	④アプリケーション実行ログフォーマットを取り決め出力ログの集計・解析ツールを作成する。 **●工夫・対応の効果** ・多量に出力されたログファイルのデータ集計作業を効率化し、エラーやボトルネック等の原因究明のためのトレーサビリティ(トランザクションの識別や問題箇所の特定)が向上する。 ・ツールを使用することにより解析精度の向上も図れる。			○	○		○
				⑤トレース機能は性能に影響を与えやすいため、トレースレベルを複数設定可能なようにし、状況に応じてレベルを変更できるようにする。						○
				⑥ソフトウェア資産の貸出/返却/更新状況を管理する。 **●工夫・対応の効果** いつ、誰が更新したのかの履歴を残し、その履歴を参照することで、ソフトウェア資産へのアクセス状況をトレース可能とする。			○	○		
	変更性 保守効率(欠陥対処や改良開発など)に関する要求水準を保全するために講じる事例	変更履歴、構成管理などの変更制御に関する工夫		①変更履歴を構成管理ツール、実装コード中、及び、テストコードに埋めておく。 **●工夫・対応の効果** 障害発生時にどの改修によるものか特定できるため、障害原因の横並び展開(調査)が容易に行える。					○	
			34－②	②機能検証確定(結合テストなど)以降の資産管理運用(市販ツール利用)による変更ルールを徹底する。ソフトウェア製品による、案件及び資産の一元管理もルール化し、案件と資産の貸出・変更履歴、リリース履歴など蓄積する。 **●工夫・対応の効果** ・パッケージを使用しているため、導入までの期間短縮が図れる。 ・障害発生時にどの改修によるものか特定できるため、障害原因の横並び展開(調査)が容易に行える。					○	
				③設計書の改定時には、"修正履歴"を必須とする。修正履歴の記述内容は標準化しており、開発部門内で標準化のチェックを行うとともに、管理部門による検収でサンプルチェックを行う。 設計書を改定せずにプログラムは改定できないルールとしており、設計書とプログラムの整合性は保っている。 **●工夫・対応の効果** 設計書により改定箇所が明らかとなるため、影響範囲特定のための工数の削減が図れる。			○			○
				④開発時は、本番データをトレースしたテストを可能とする。 **●工夫・対応の効果** 本番入力データを再現したテストを行うことにより、実運用下でのシミュレーションが可能となり、より精度の高い製品の提供が可能となる。					○	

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
保守性	変更性 保守効率(欠陥対処や改良開発など)に関する要求水準を保全するために講じる事例	母体システムの構造化度、変更生産性など変更容易性向上に関する施策		①コーディング規約を規定する。 ・サンプルプログラムを1本作成し、全プログラマがサンプルプログラムをベースに各業務別のプログラムをコーディングする。 ・コーディング規約においては、以下の点を規定し構造化を図る。 ・初期処理用関数名、業務処理用関数名、終了処理用関数名、異常処理用関数名、共通処理用関数名の命名規則。 ・関数の中で別の関数を呼び出すネストを最大3段階までに規定する。 **●工夫・対応の効果** 全プログラムが規約に従った構造になっているため、どのプログラムを改修するにも大枠での理解が早い。				○		
			35－②	②アプリ基盤のアーキテクチャとして、システム(開発資産)を階層的に構造化(プレゼン、サービス、モデル、データソース)して、かつ階層間のインタフェースを標準化する。 **●工夫・対応の効果** システムの経年劣化を防止し変更に強い資産となる。			○			
				③影響分析用リポジトリの作成と影響調査を市販製品にて実現する。生産性向上。変更品質の向上。 **●工夫・対応の効果** パッケージを使用しているため、導入までの期間短縮が図れ、生産性や変更品質の向上が見込まれる。			○			
				④どの段階でサブルーチンを呼んだか、DBからどのようなデータをアクセスしたか等々の情報を収集する。 **●工夫・対応の効果** 実際の動きをトレースすることにより、プログラムの動作が明らかとなるため、修正箇所の特定に役立つ。			○			
		保守ドキュメント、コメント率など変更容易性向上に関する施策		①JOBよりJOBフローを自動作成するツールを使用し、毎夜間処理にてJOBフローを自動作成する。 **●工夫・対応の効果** ファイルの接続状況等確認が容易になり、調査精度の向上、調査工数の削減に寄与する。				○		
				②ソースプログラムを解析して、CALL／CALLER関係から階層構造図を作るツールを作成する。 **●工夫・対応の効果** プログラムの関連が明確になるため、保守、改修工数の削減に寄与する。				○		
			17－④	③プログラムソースへの修正履歴等のコメント挿入を標準化する。設計書を保守ドキュメントとしても使用可能となるよう考慮しながら、補足事項等記載内容を充実させる。 **●工夫・対応の効果** これまでの変更内容が明確になることで、改良開発での調査、変更作業がしやすくなる。				○		
				④コメント率は50%程度(実行行数と同程度のコメント行)をコーディング規約に規定し、計測・評価する。加えて、コメント率を解析するための仕組み(ツール)を作成する(市販の解析ツール等で十分解析可能)。 **●工夫・対応の効果** コメントとして処理内容を記載することで、設計書としての利用ができ、保守、改修工数の削減に寄与する。				○		
				⑤ドキュメント生成ソフトウェアの規約に則った、アプリケーション作成規約を作成し、ドキュメンテーション製品を適用する。また、変更資産と関連ドキュメントのリプリケーション管理を行う。 **●工夫・対応の効果** 保守用ドキュメントを自動生成するため、コスト削減が図れるとともに、調査工数の削減に寄与する。				○		
	安定性 保守作業によって発生する可能性のある[故障・誤り混入＝デグレード]に関する要求水準を保全するめに講じる事例	保守作業での欠陥混入是正に関する工夫	24－②	①システムの改良等で発生したテストケース・データを累積しておき、次回のリグレッションテスト実施時に追加する。 **●工夫・対応の効果** 改修に伴う箇所だけでなく、少ない工数で、プログラム全体に関してテストを実施することができるため、改修箇所が及ぼす影響を早期に把握することができる。					○	
				②デグレード確認のために新旧プログラムによる実行結果を比較し、修正対象以外に影響がないことを確認する。 **●工夫・対応の効果** デグレードによる障害の発生を未然に防止することができる。				○	○	
				③結合テストを自動化する仕組み(予想結果と突合せOK／NGのリスト作成)を構築する(Excelマクロで構築し、テストケースの追加／変更や、出力結果(OK／NG)もExcel上で出来ている)。 ・バージョンアップ作業前に、システム、データのバックアップを取得する。 **●工夫・対応の効果** 障害対応した後は、毎回、開発時のテストを全件(約500項目)実行してデグレードを防止する。				○	○	
				③"他へ影響を与えないことを確認した手段・保証する内容"を開発担当に具体的に記載させ、管理部門で承認した上でリリースするルールを定める。ごく小規模な改定や保守作業においても使用する定型のフォームに上記内容を記入し全社で使用する。 **●工夫・対応の効果** ・影響範囲の特定方法を公にすることにより、客観的な判断を行うことができる。 ・デグレードによる障害の発生を未然に防止することができる。					○	○
				④自動生成するソースコード資産と手組作成するソースコード資産が完全に分離されるようにシステム(開発資産)を構造化する。技術的に可能な部分に関しては、設計ドキュメントからソースコードを自動生成する機構を装備して設計情報と実装資産が乖離しない工夫をする。 保守、運用においては以下の対応を行う。 ・影響検索システムにより影響漏れを摘出する。 ・変更要件と変更内容の第3者を含めたレビューを実施する。 ・修正における機能テスト範囲の明確化とテスト結果の第3者レビューを実施する。 ・本番環境(あるいは、ほぼ本番環境)での影響範囲最終テストを実施する。 ・リリースの切り戻し手順の確立と、それらを含むリリース作業の訓練を実施する。 **●工夫・対応の効果** ・影響範囲の特定方法を公にすることにより、客観的な妥当性の判断を行うことができる。 ・デグレードによる障害の発生を未然に防止することができる。				○	○	
				⑤・複数メンバによるダブルチェック(指差し確認)を実施する。 **●工夫・対応の効果** 障害の発生を未然に防止することができる。					○	
		母体品質向上施策		①不要となった資産は、稼働サーバから削除しない。 **●工夫・対応の効果** 資産が増加しメンテナンス効率が落ちる原因となるが、資産を削除しないため、障害発生時早期に復旧することができる。						○
			17－⑤	②改良開発でのテストフェーズにおいて、テストケースに過去本番障害事例を追加する。過去障害時に解析、検証で利用したデータも保管しておき、デグレード確認テストとして利用する。 **●工夫・対応の効果** 潜在欠陥の摘出に効果があり、母体品質向上に寄与する。				○		

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
保守性	安定性 保守作業によって発生する可能性のある[故障・誤り混入＝デグレード]に関する要求水準を保全するために講じる事例	システム保有データの整合性維持に関する施策		①サブシステム内の「主管DB整合性確認ツール」を作成し、結合テスト以降の検証に利用する。さらにDB変更時には当該ツールも併せて変更し維持する。 **●工夫・対応の効果** ・本ツールにて項目の取り得る値・設定箇所をシステム的に検証することにより不整合箇所が機械判定されるため、人手による検証漏れを抑え、サブシステム内での障害摘出漏れを防止できる。					○	
			36－①	②DBの相関関係やデータの整合性を検証するツールを開発し、通常運用や障害対応のなかで走行させる。システム内やシステム間でのDB整合性をチェックし、チェック結果を出力する。 **●工夫・対応の効果** ・当該ツールを実行することで、表面化しにくいDB間の不整合を早期に摘出することが把握できる。 ・障害の早期解決(原因の早期特定)に寄与する。 ・テスト工程に適用することで、テスト品質向上(摘出漏れ防止)および生産性向上も期待できる。		○			○	○
		バージョンアップによるデグレード防止に関する施策		①ある日の本番ログからトランザクションを生成し、バージョンアップ前後の本番擬似環境に投入し、全DB、ファイルを物理的にコンペアする仕組みを構築する。トランザクションの順序性を維持する場合は、トランザクションをシリアルに投入することとなり、投入可能なトランザクション数を制限する必要がある。この場合は、網羅性を考慮して投入するトランザクションを選択する。 **●工夫・対応の効果** 本番データにて実行テストを実施することにより、稼働前に動作結果を把握でき、障害の発生を未然に防止することができる。					○	○
				②本番での新旧システム並行ラン期間を設定し、データ処理結果の新旧コンペアテストを実施する。 **●工夫・対応の効果** ・デグレードによる障害の発生を未然に防止することができる。 ・新システムにて障害が発生しても、旧システムが動作しているため、障害の影響を最小限に留めることができる。					○	
			37－①	③OSやミドルウエア等のアップグレードにおいて、明示的な非互換項目が無い場合でも、稼働確認テストに加え開発環境で十分な期間(2～3ヶ月程度)稼働させスタビリティを確認してから本番機へ適用する。 **●工夫・対応の効果** 試行環境にて稼働することで、動作結果を事前に把握でき、障害の発生を未然に防止することができる。					○	
				④バージョンアップ作業前に、必ずシステム、データのバックアップを取得する。 **●工夫・対応の効果** 作業に問題が発生した際に、短時間で戻すことができ、影響の拡大を未然に防ぐことができる。					○	
				⑤原則(運用に支障をきたさない限り)ソフトウェアのバージョンアップは実施しない。また、バージョンアップ時は試行環境を構築して、事前検証を実施する。 **●工夫・対応の効果** ・デグレードによる障害の発生を防止することができる。 ・試行環境にて稼働することで、動作結果を事前に把握でき、障害の発生を未然に防止することができる。					○	
			34－①	⑥ソフトウェアのバージョンアップ(本番移行)について、厳格な運用規準を設ける。本番障害を除き、事前の移行申請－承認を経て試行環境からのみ移行可能とする。試行環境での稼働実績(品質「デグレード防止」に関する結果承認)がない限り、本番移行することができない仕組みとする。 **●工夫・対応の効果** ・デグレードによる障害の発生を防止することができる。 ・試行環境にて稼働することで、動作結果を事前に把握でき、障害の発生を未然に防止することができる。					○	
		障害混入確率の低減に関する施策		①アプリケーションソフトは機能毎にプログラムの差し替えを可能にする。 **●工夫・対応の効果** システムの影響を最小限に留めることにより、障害発生時の調査工数の削減に効果があるとともに、障害の拡大を未然に防止することができる。			○	○		
			35－①	②過去の障害混入事例の再発防止観点をリストアップしたチェックリストを励行する。 **●工夫・対応の効果** 類似障害の発生を防止することにより、障害件数の削減が期待できる。						○
				③システムの変更(障害対応、機能追加/変更)は可能な限り、まとめて取り込む(発生の都度頻繁にリリースしない)。リリース前の検証手順を強化し、変更～リリース手順(承認～実施)を遵守する。 **●工夫・対応の効果** ・リリース回数を減らすことにより、本番対応工数の削減が期待できる。 ・リリース手順を守ることで、予期せぬ障害の混入を未然に防止することができる。						○
	試験性 保守に必要となるテストに要する労力に関する要求水準を保全するために講じる事例	本番リリースを保証するためのテスト範囲特定方法などに関する取り組み	9－② 21－⑤	①本番(同等)環境でテストする項目を用意して、テストツールにより予想結果と一致することを確認してからリリースする。 **●工夫・対応の効果** 常に実施すべき確認項目を予め提供することで、一定品質を保つことができ、障害の発生を未然に防止することができる。					○	
				②開発/改修対象範囲を調査するため、プログラム、ファイルのSCANツールを使用し、SCAN結果をユーザとレビューしたうえで、開発/改修対象範囲を確定する。 **●工夫・対応の効果** ・影響範囲を特定できるため、改修箇所、テスト範囲が明確となり、テスト漏れによる障害の発生を未然に防止することができる。 ・テスト範囲については、データ接続先に重要な処理がある場合、修正対象以外でも実施するようにし、重大障害の発生を未然に防止する。					○	
				③開発資産の構成管理機能の1つとして、データ項目名やプログラム名をキーに関連する資産(設計ドキュメントやプログラムなど)を検索できる仕組みを提供する(影響検索ツール)。 **●工夫・対応の効果** 影響範囲を特定できるため、改修箇所、テスト範囲が明確となり、テスト漏れによる障害の発生を未然に防止することができる。					○	○
			9－② 21－⑤	④本番環境とテスト環境の差異を明確化した上でテスを実施する。 **●工夫・対応の効果** 本番環境とテスト環境との差異を明確にすることにより、本番環境でしか確認できない項目の扱いについて、ユーザと事前に調整することができ、対象項目に関わる箇所を重点的にレビューする等、障害除去に向けた早期の対応を行うことができる。					○	○
			22－②	⑤本番リリース前に、本番と同等のハードウェア環境下での負荷テストが実施できるよう、テスト用データベースやトランザクションデータを準備する。高負荷テストでのパフォーマンス測定結果(目標達成度合い)をリリース条件とする。 **●工夫・対応の効果** 常に実施すべき確認項目を予め提供することで、一定品質を保つことができ、障害の発生を未然に防止することができる。					○	○

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性（2次）	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
保守性		保守テストでのテストツール（自動再帰テストツールなど）、本番環境具備、標準テストセットの具備などのテスト環境整備に関する工夫	38－①	①結合テスト用（サブシステム内結合用）に「DB生成ツール」を作成し、保守案件にて変更が発生する場合はツールもメンテナンスする。 **●工夫・対応の効果** 本ツールにより、人手によるデータ準備工数が大幅に削減され、かつ、保守のテストにおいてデータ整備不備による手戻りの発生を防止することができる。					○	
				②データベースSQLコーディングチェックプログラムを作成しコーディング規約チェックやアクセスプラン解析等SQL品質向上に役立てる。 **●工夫・対応の効果** SQLの保守可読性や性能等の品質向上に寄与する。				○		
				③Web系システムでのトランザクションデータを自動採取し、自動再実行（リグレッションテスト）に利用する。 **●工夫・対応の効果** 実際の入力データを使用するため、テストデータの作成工数の削減に繋がるとともに、きめ細かなテストを実施することができ、開発工数の削減に寄与する。					○	
				④アプリケーション、JCLが意識する環境（ファイル名、ファイルサイズ等）は、本番と開発環境で同じ構成とする（環境を意識して変更を加える必要がない）。 **●工夫・対応の効果** テスト環境用JCL、ファイルを本番環境用JCL、ファイルに置き換える必要がなくなるため、工数の削減に繋がるとともに、人手を介することによる障害の発生を未然に防止することができる。					○	
		試験性に配慮したソフトウェアおよびデータの構造化、ならびに他システム接続方式に関する施策	5－③	①アプリ基盤として、EJBコンテナ、DBマネジメントシステムなどに依存しないソフトウェアの構造化を行い、PCなど簡易・軽量な試験環境で十分なテストを実施する。 **●工夫・対応の効果** テスト環境を意識した設計を行うことにより、多くのプログラムのテストをPC環境等にて実施できるため、他者との競合による手戻り等の発生を防ぐことができ、テスト密度を向上させることができる。					○	
				②設計、製作時から試験計画（試験支援ツールなど）を立案し、かつ、設計レビューにて試験担当者がレビューに参加する。 **●工夫・対応の効果** 設計段階から試験工程を意識することにより、試験工程に必要な資源、プロトタイプによる試験の実施等、開発工程と並行して検討することができるため、開発総工数の削減に繋がる。			○			
				③データのフォーマットの決定権を有する場合には、一般的に流通している表計算ソフトウェア等で作成可能な、視認性の高いフォーマットに限定する。 **●工夫・対応の効果** ダウンロードする際に、コード変換等不要となるため、テスト検証工数の削減に繋がる。			○			
	保守性標準適合性	保守性に対応する規定（業務、内部統制、ISMS、国際会計など）および開発標準の適合に関する監査対策		①ITサービスに関する分野の規格・基準・ガイドライン等。 ・ISO/IEC20000-1:2005、ISO/IEC20000-2:2005、PD0005、PD0015、ITIL。 ・ISO/IEC9126(JIS X0129)：ソフトウェア品質特性。 **●工夫・対応の効果** 規格より、ソフトウェアの保守性に影響を与える項目をチェックリストに整理し、運用することでコスト削減を図る。			②			○
			39－①	②システム毎に当該システムの概要を記載したプロファイルを作成し、システム構成や関連システム、非機能要求レベル、障害時の影響レベルなどを規定することを標準化する。当該プロファイルは保守ドキュメントの位置づけであるが、リスク分析の元ネタとして活用し、保守コスト評価にも利用する。 **●工夫・対応の効果** 一元化されたプロファイルを参照することにより、システムの状況が容易に理解でき、保守工数の低減に寄与する。			○			
移植性	環境適応性 対象のソフトウェアを複数の動作環境で稼働させる要求水準を保全するめに講じる事例	ソフトウェア（アプリケーション）の環境適応性向上施策（OS、ミドルウェア、DBマネジメントシステムなど）		①業界標準製品や言語を採用する。 **●工夫・対応の効果** 開発経験者が多く、人員の手当てが行いやすく、価格交渉にて優位に立ちやすい。			○			
			37－②	②（開発）プラットフォームに依存しないJavaの特性を活かし、JavaベースのSIを推進するとともに、アプリ基盤としてEJBコンテナやDBマネジメントシステムなどに依存しない構造化を図る。 **●工夫・対応の効果** ・開発経験者が多く、人員の手当てが行いやすく、価格交渉にて優位に立ちやすい。 ・アプリ基盤にて難解な箇所を吸収しているため、経験の浅い要員も開発できる環境となる。			○			
	設置性 対象のソフトウェアを移植する際に必要となる労力に関する要求水準を保全するめに講じる事例	移植時の設置作業の確実性向上施策（移植作業支援ツール（移植箇所特定）など）		①同一サーバ内に環境変数定義を複数作成し、環境変数を資産により使い分ける環境を構築する。 **●工夫・対応の効果** 環境による違いを環境変数にて吸収しているため、移植時プログラムでの対応が不要となり、コスト削減に寄与する。						○
	共存性 共通の資源を共有する環境の中で、他のソフトウェアとの共存に関する要求水準を保全するために講じる事例	同一環境下（同一サーバ、同一DBマネジメントシステムインスタンスなど）で、複数アプリケーションを共存させるための施策		①仮想化技術、スケールアップ／スケールアウト構成などを用いて、能力アップ時には、アプリケーションの変更が発生しない、あるいはシステム停止を伴わない構造を採用する。 **●工夫・対応の効果** アプリケーションの変更が伴わないため、スケールアップ/スケールアウトを容易に行うことができるため、性能向上要求に迅速に対応することができる。			○			
				②同一環境下（同一サーバ、同一DBマネジメントシステム・インスタンスなど）で、複数アプリケーションを共存させるために、環境IDという論理層により独立性と並行運用性を実現する仕組みを用意する。 ・同一サーバ内にトップディレクトリのみ分けた同一環境を開発環境分（単体、結合など）作成する。 ・同一サーバ内に環境変数定義を複数作成し、環境変数を資産により使い分ける環境を構築する。 **●工夫・対応の効果** 環境による違いを環境変数等にて吸収しているため、プログラムでの環境を意識する必要がなく			○			
				③海外システムは、1ホストで複数拠点を処理する必要があり、しかも時差があるので、オンライン開閉局・バッチ開始終了を制御する仕組みを考える必要がある。アプリケーションが意識しないよう開閉局管理を共通機能化した。			○			

代用特性を利用したシステムの信頼性向上への工夫事例

特性	副特性	代用特性(2次)	障害・再発防止事例対応No.	工夫事例	工夫事例対象工程					
					契約	要件	設計	実装	テスト	保守運用
移植性	置換性 ある環境で対象のソフトウェアと置き換えて他のソフトウェアを使用する場合の労力に関する要求水準を保全するために講じる事例	制度変更などシステム機能の置換をし易くするための施策(部品化など)		①影響度分析を考慮して、変数名、メソッド名等を同一にする。 **●工夫・対応の効果** 影響調査範囲の特定が容易となり、解析性・変更性が向上するとともに改修工数の削減にも寄与する。				○		
				②固定値をプログラム内に保持するのではなく、定数表としてサブルーチン化しておき、アプリケーションには影響を与えない仕組みとする。定数表は開始/終了日を保持する形式としておくことで事前のリリースを可能とする。 **●工夫・対応の効果** 日付の変更、値の変更時にサブルーチンを修正するのみで機能提供を実現できるため、改修箇所も限定され、品質の安定に繋がる。また改修コストの削減にも寄与する。				○		
				③運賃適用ルールをDBに保有して、プログラム修正なしに運賃改定する仕組みを保持。ルールの事例として下記を制定する。 ・運賃毎に取り扱える販路(Web、代理店. . .)を限定する。 ・割引系運賃の適用条件(搭乗何日前だったら購入できる)。 ・株主優待券やマイルによる割引ルール　等。 ・その他制度やルールのデータ化。 **●工夫・対応の効果** 機能変更時、プログラムの修正を伴わないため、品質の安定が見込まれる。ただし、DBの変更を伴うため、事前にテストできる環境の構築が必要となる。				○		
	移植性標準適合性	移植性に対応する規定(業務、内部統制、ISMS、国際会計など)および開発標準の適合に関する監査対策		①ISO/IEC9126(JIS X0129):ソフトウェア品質特性。 **●工夫・対応の効果** 規格より、ソフトウェアの品質特性に影響を与える項目をチェックリストに整理し、運用することでコスト削減が図れる。						○