

システム要件定義プロセスガイド

第 1.10 版

2018 年 08 月 29 日

TIS 株式会社



この 作品 は [クリエイティブ・コモンズ 表示 - 継承 4.0 国際 ライセンス](https://creativecommons.org/licenses/by-sa/4.0/) の下に提供されています。

變更履歷

[illegible]

目次

S1 システム要求の収集と整理.....	4
S1-01 現行システムの調査.....	4
S1-01-01 現行システムの調査.....	5
S1-02 課題の抽出と原因分析.....	7
S1-02-01 システム化企画資料からの課題抽出.....	8
S1-02-02 業務要件定義の申し送り事項からの課題抽出.....	9
S1-02-03 現行システムからの課題抽出.....	10
S1-02-04 ステークホルダーからの課題抽出.....	11
S1-02-05 課題の原因分析.....	12
S1-03 課題解決の実現手段検討.....	14
S1-03-01 課題解決後の状態設定.....	15
S1-03-02 実現手段の方向性検討.....	16
S1-03-03 実現手段の詳細化.....	17
S1-04 機能の整理.....	18
S1-04-01 システム要求のリストアップ.....	19
S1-04-02 機能の整理.....	20
S1-04-03 機能間の整合性、一貫性の確認.....	22
S1-04-04 機能のステークホルダー確認.....	23
S2 機能要件の定義.....	24
S2-01 機能要件の定義.....	24
S2-01-01 画面機能の要件定義.....	25
S2-01-02 帳票機能の要件定義.....	27
S2-01-03 外部 IF 機能の要件定義.....	28
S2-01-04 バッチ機能の要件定義.....	30
S2-01-05 論理データモデルの定義.....	31
S3 非機能要件の定義.....	33
S3-01 モデルシステムの選定.....	33
S3-01-01 モデルシステムの選定.....	34
S3-02 可用性要件の定義.....	36
S3-02-01 業務継続性.....	37
S3-02-02 目標復旧水準.....	38
S3-02-03 稼働率.....	39
S3-02-04 災害対策.....	40
S3-02-05 耐障害性.....	41
S3-03 性能・拡張性要件の定義.....	42
S3-03-01 業務処理量.....	43
S3-03-02 性能目標値.....	44
S3-03-03 リソース拡張性.....	45
S3-03-04 性能品質保証.....	46
S3-04 運用・保守要件の定義.....	47
S3-04-01 保守運用.....	48
S3-04-02 障害時運用.....	50
S3-04-03 運用環境.....	51
S3-04-04 サポート体制.....	53
S3-04-05 その他の運用管理方針.....	55
S3-05 運用監視要件の定義.....	56
S3-05-01 監視情報.....	57
S3-05-02 ソフトウェアに対する監視.....	58
S3-05-03 ハードウェアに対する監視.....	59
S3-05-04 ネットワークの監視.....	60
S3-06 バックアップ要件の定義.....	61
S3-06-01 データ復旧範囲.....	62
S3-06-02 バックアップ利用範囲.....	63
S3-06-03 バックアップ自動化の範囲.....	64
S3-06-04 バックアップ取得間隔.....	65
S3-06-05 バックアップ保存期間.....	66
S3-06-06 バックアップ方式.....	67
S3-07 ネットワーク要件の定義.....	68
S3-07-01 既存回線の利用有無.....	69
S3-07-02 WAN 構成.....	70
S3-07-03 LAN 構成.....	71

S3-07-04	外部システム接続構成	72
S3-07-05	保守回線	73
S3-08	移行要件の定義	74
S3-08-01	移行時期	75
S3-08-02	移行方式	76
S3-08-03	移行対象	77
S3-08-04	移行計画	79
S3-09	セキュリティ要件の定義	81
S3-09-01	情報セキュリティに関するコンプライアンス	82
S3-09-02	セキュリティリスク分析	83
S3-09-03	セキュリティ診断	84
S3-09-04	認証機能	85
S3-09-05	利用制限	86
S3-09-06	データ暗号化	87
S3-09-07	不正監視	88
S3-09-08	ネットワーク対策	89
S3-09-09	マルウェア対策	90
S3-09-10	Web 実装対策	91
S3-10	システム環境	92
S3-10-01	接続対象	93
S3-10-02	稼働環境	94
S3-10-03	特定製品指定有無	95
S3-10-04	耐震/免震レベル	96
S3-11	テスト要件の定義	97
S3-11-01	テスト工程定義と各テスト工程の役割分担	98
S3-11-02	各テスト工程の品質管理	101
S3-12	非機能要件の制約条件の定義	102
S3-12-01	非機能要件の制約条件の定義	103
S3-13	非機能要件の対応レベル決定	104
S3-13-01	対応レベルの補正と決定	105
S4	全体要件の精査	106
S4-01	要件の検証と妥当性確認	106
S4-01-01	機能要件と非機能要件の検証	107
S4-01-02	機能要件と非機能要件の妥当性確認	108
S5	全体要件の合意と承認	109
S5-01	要件の実施対象決定	109
S5-01-01	工数、コスト試算	110
S5-01-02	スケジュール試算	111
S5-01-03	実施対象の決定	112
S5-02	要件定義書の完成	114
S5-02-01	要件定義書の完成	115
S5-03	要件の合意と承認	116
S5-03-01	お客さまによる要件の合意と承認	117
S6	引継ぎ	118
S6-01	設計工程への引継ぎ	118
S6-01-01	設計工程への送り事項整理	119
S6-01-02	設計工程への引継ぎ	120

【添付資料】

1. システム要件定義プロセス一覧
2. システム要件定義プロセスフロー

【留意事項】

要件定義フレームワークは、お客さまとベンダーの協働で要件定義を実施することを想定し、ベンダーの視点から要件定義の進め方をガイドしています。この場合、お客さまは要件定義の主体者として課題や要求の抽出、要件を決定する立場となり、ベンダーはそれらの活動を支援する立場となります。ユーザー企業自身で要件定義を実施する場合は、要件定義フレームワークに記述された「お客さま」を「自社」に、「ベンダー」をユーザー企業内で要件定義推進を担う組織・人(例えば、情報システム部門)に読み替えてご利用ください。

S1 システム要求の収集と整理

S1-01 現行システムの調査

サブプロセス目的
システム課題・システム要求を正確に把握するために、システムの現状を把握する。
サブプロセス概要
画面機能・帳票機能・バッチ機能・外部 IF 機能・データモデル・インフラ・運用設計について記載された資料をもとに、現状を調査する。

S1-01-01 現行システムの調査

アクティビティ概要	
<p>現行システムを調査する。機能の観点では、システム機能一覧をもとに、各画面や帳票などの構成やレイアウト、項目レベルでの調査を行う。非機能の観点では、現状のハードウェア、ミドルウェア、ネットワーク構成図や、運用マニュアルをもとに調査を行う。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要件定義成果物 ※システム機能一覧(As-Is)、CRUD(As-Is) ● 現行システムの資料(As-Is) ※外部 I/F 設計(As-Is)、画面設計(As-Is)、帳票設計(As-Is)、 バッチ設計(As-Is)、データモデル(As-Is)、 現行システム機能俯瞰図、ソフトウェア構成図、 ミドルウェア構成図、現行ネットワーク図、 現行サブシステム構成図、現行サブシステム間 I/F、 現行開発環境マニュアル、現行試験環境マニュアル、 現行運用保守契約書、現行運用マニュアル、 現行システム運用スケジュール、現行保守マニュアル、 現行オペレーションマニュアル、現行リモート接続手順書、 現行定期報告会資料、現行監視設定書、現行監視手順書、 拠点内・アプリ別利用者数、端末台数 	<ul style="list-style-type: none"> ● 現行システムの資料(As-Is) ※必要な現行システム情報に合わせて、左記インプットに指定された資料から必要なものを選択し、調査結果をまとめる。
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>全体</p> <hr/> <ul style="list-style-type: none"> ➤ 現行調査結果をシステム要件検討でうまく活用するために、全体像と詳細調査結果の関連、必要な調査内容を明確にする。 現行システム調査は、トップダウンとボトムアップの両方の観点で調査対象、内容を明確にする。 現行システム機能俯瞰図、ソフトウェア構成図などによりシステム全体の概観を捉えた後、それらとの関連を明確にしながら詳細調査を進める。 また、その結果に応じて現行システム機能俯瞰図等の不備、不足を随時訂正する。 <p>外部システム</p> <hr/> <ul style="list-style-type: none"> ➤ 非機能要件を検討するために、連携する社内外の外部システムに関する情報が必要である。 外部システム側の運用体制やサービスレベルなどが、新システムの可用性、運用要件に影響する可能性がある。 例えば、外部システムに影響を与えないようにするために、構築対象システムの可用性の対応レベルを外部システムの可用性の対応レベルまで引き上げるなどの検討が必要になる場合がある。 <p>非機能要件に関する現状調査</p> <hr/> <ul style="list-style-type: none"> ➤ 運用資料や定期報告会資料から、業務処理量とその変動傾向や要因、リソース使用状況、バッチ処理運用実績等を確認する。 非機能要件に影響する業務特性を理解できる。 	
上手く進めるためのポイント、注意事項	
<p>全体</p> <hr/> <ul style="list-style-type: none"> ➤ 設計書を調査するだけでなく、現行システムを実際に操作、利用することで、現状理解を深めることができる。 また、現行システム設計書の最新化状況を確認できる。 ➤ 現行システムのソースコードから、各機能の規模感を詳細に把握できる。 ➤ 現行システムの調査結果を、システムの機能要件の検討時に有効活用できるようにするため、業務要件定義で作成したシステム機能一覧(As-Is)と、現行システムの調査結果の外レサビリティを確保する。 この過程で、業務要件定義で作成したシステム機能一覧(As-Is)にシステム機能の漏れや不要なシステム機能が含まれているなどの誤りがあった場 	

合は、業務要件定義に誤りがある可能性があるため、影響調査を行う。

- 現行機能がサブシステム構成上どのように配置されているかを確認し、理解しておくこと。
新システムの機能追加による影響調査に必要な。

画面

- 現行システムの資料から把握できない、手運用で補っているシステム要求候補やその他システム課題の把握に、お客さまの業務観察や利用内容実態のヒアリングが有効である。

IF 機能

- 外部 IF ファイルなどの出力データを調査する場合は、設計書確認に加えて、実際のデータを確認すると良い。
設計書に未記載のデータ様式や、手運用で補っているデータルール(例:備考データの先頭に顧客ステータスを手入力する)等を確認でき、システム要求検討のインプットになる。

データ

- 実データの調査結果から、設計と実装との間に乖離がないことを確認する。
例えば、データベースに登録されている実データを分析し、コード設計に記載されていないコード値がないかの確認や、論理データ設計に記載されていないエンティティや属性がないかを確認する。

非機能要件に関する現状調査

- 非機能要件に関する企業標準(監視標準、UI 標準、ネーミング規約、セキュリティ対策方針など)の有無をお客さまに確認し、存在する場合は内容を確認する。
RFP などに制約条件として定義されていない場合でも、企業標準に準拠することが暗黙のルールとなっている可能性もあるため、企業標準の有無、準拠の必要性、およびその内容を確認する必要がある。

適用技法

- (特になし)

S1-02 課題の抽出と原因分析

サブプロセス目的
プロジェクトの目的・目標および業務要件を達成するために、解決すべきシステム上の課題を明確にする。 また、明確にしたシステム課題に対して適切な実現手段を検討するために、システム課題の根本原因を特定する。
サブプロセス概要
システム要件で扱う課題とは、プロジェクトの目的・目標および業務要件を満たすために IT システムが解決すべき課題である。 例えば、システム機能のレスポンスに対する課題などについては、業務要件定義では取り扱わず、システム要件定義で取り扱う。 このシステムが抱える課題を、システム化企画資料、業務要件定義の申し送り事項、現行システムの調査結果、ステークホルダーへのインタビュー等によって抽出する。抽出したシステム課題を取りまとめ、原因分析を行う。

S1-02-01 システム化企画資料からの課題抽出

アクティビティ概要	
<p>業務要件実現のために解決すべきシステム課題や、システム化方針に着目して、システム課題を抽出する。 例えば、以下のように、「内部統制対応にあたって、現行システムが抱えるセキュリティへの課題がないか」や、「〇〇入力業務の生産性向上にあたって、現行画面のユーザービリティに関わる課題がないか」など、業務要件実現の観点から、システム側で解決すべき課題を抽出する。</p>	
課題 ID	課題名
SP001	内部統制に当たり各情報にアクセスできるユーザーの権限の適正化
SP002	〇〇入力業務の省力化、効率化
インプット	アウトプット
<ul style="list-style-type: none"> システム化企画資料 RFP 	<ul style="list-style-type: none"> システム課題一覧
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ システム化企画資料、RFP に記載された課題への対策は、理想や仮定で記載したものがあり、課題の妥当性、実現手段の実現性や他との整合性等が未検討の場合があることに注意する。 記載された課題と対策を精査せずにシステム課題一覧に取り込むと、後続作業で不適切なシステム要求が導出されて作業手戻りが発生する、不適切なシステム要件が実装されて投資対効果が低下する、等の問題が生じる場合がある。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ システム化企画資料、RFP から、①現状の問題点、②問題点の原因、③あるべき姿(問題解決時に実現される状態)、④解決策、を抽出し、それらが論理的に関係づけられた状態で、システム課題一覧にまとめる。 システム化企画資料、RFP に②から④が網羅的かつ体系的に書かれていない場合もあるが、後続アクティビティで原因分析や実現手段検討を行う際の参考情報として、把握できた範囲でシステム課題一覧にまとめる。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S1-02-02 業務要件定義の申し送り事項からの課題抽出

アクティビティ概要	
業務要件定義から、システム要件定義へ申し送りされたシステム課題(※)を抽出する。下表の下線部のような課題が抽出される。	
課題 ID	課題名
SP001	内部統制にあたり各情報にアクセスできるユーザーの権限を適正化
SP002	〇〇入力業務の省力化、効率化
SP003	<u>△△業務で参照している帳票の出力に時間が掛かる。</u>
SP004	<u>△△業務で参照している帳票は、前日の集計情報の為、実態と異なる場合がある。</u>
※：業務課題に混在して抽出されたシステム課題は、業務要件定義では検討せず、システム要件定義へ申し送りしている。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要件定義書 ● システム要求一覧(業務要件定義時) ● 申し送り事項一覧 	<ul style="list-style-type: none"> ● システム課題一覧
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
➤ 申し送りされたシステム課題は詳細化が不足している(理由は上記※参照)ため、経緯と背景を業務要件定義書やその元となった資料で確認し、お客さまと共同で詳細化する。	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S1-02-03 現行システムからの課題抽出

アクティビティ概要		
システム化企画資料や業務要件定義で抽出したシステム課題やシステム化方針について、「S1-01 現行システムの調査」で把握した現行システムと照らし合わせ、具体的なシステム構造上の問題点や運用上の問題点として明らかにする。		
インプット		アウトプット
<ul style="list-style-type: none">● システム課題一覧● 業務要件定義成果物● 現行システムの資料(As-Is)		<ul style="list-style-type: none">● システム課題一覧
手順		
<p>1. システム課題一覧にまとめた、システム化企画資料や業務要件定義で抽出したシステム課題を、「S1-01 現行システムの調査」で把握した現行システムと照らし合わせて具体化する。</p> <p>例えば、「現行画面のユーザービリティに関わる課題」というキーワードに対して、現行システムの各画面の遷移やレイアウト、表示情報や入力情報に抱えている具体的な課題を抽出する。</p> <p>以下に課題抽出の例を挙げる。例えば、下表の下線部のような課題抽出や内容の具体化が行われる。</p>		
課題 ID	課題名	課題内容
SP001	内部統制にあたり各情報にアクセスできるユーザーの権限を適正化	<u>画面：入出力項目がどのユーザーが利用しても同じ状態になっている。</u> <u>帳票：すべての帳票が出力可能な状態となっている。</u>
...
SP005	現行の運用設計では、障害発生時に復旧目標を満たせない可能性がある。	現行システムの障害時では、日次で取得しているテープバックアップを利用して再構築することにより復旧することで合意していた。 <u>お客さま運用でバックアップ取得結果の監視を行うことになっているが、運用が徹底されていない。</u>
【重要】上手く進めるためのポイント、注意事項		
—		
上手く進めるためのポイント、注意事項		
➤ システム構造によって制約が発生している点や、運用が複雑になっている点などに注目して、現状の課題を発見する。		
適用技法		
<ul style="list-style-type: none">● （特になし）		

S1-02-04 ステークホルダーからの課題抽出

アクティビティ概要	
<p>システム課題一覧に整理したシステム課題の妥当性をステークホルダーと確認する。 また、それらを叩き台としてステークホルダーから未認識のシステム課題を抽出する。 抽出したシステム課題から、プロジェクトスコープに含め、原因分析対象とするシステム課題を選別し、ステークホルダーと合意する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム課題一覧 	<ul style="list-style-type: none"> システム課題一覧
手順	
<ol style="list-style-type: none"> S1-02-01～S1-02-03 で抽出した課題の妥当性を確認するためのヒアリング項目やアンケート項目を準備し、ステークホルダーへのインタビューやアンケートを行う。 システム化企画や業務要件定義では抽出しにくい、現行システムの利用者や運用者視点のシステム課題を抽出するために、ヒアリング項目やアンケート項目を準備し、ステークホルダーにインタビューやアンケートを行う。 インタビューやアンケートの結果をもとにして、課題の見直しを行う。 解決による投資対効果が大きい課題にプロジェクトを注力させるために、プロジェクトの目的、目標に対する効果や緊急度等の観点から、プロジェクトスコープに含め原因分析対象とする課題を、ステークホルダーと合意する。 	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ステークホルダーから抽出した課題には、抽象的な課題も多い。 例えば、「xxx が使い物にならない。」のような課題では、何が悪いのかがわからない。 そのような場合は、ステークホルダーとともに、根本原因が検討できるレベルに課題を具体化する。 適用技法に記載された技法の効果を高めるには、現在の担当や職制でインタビューやアンケートの対象を選ぶだけでなく、検討対象となる業務範囲、システム化機能範囲においてキーパーソンとなるステークホルダーの参加が得られていることが重要となる。 要件定義計画プロセスの「C1-02 ステークホルダーの定義」で分析した結果を参考にして、対象を選定する。 プロジェクトの目的・目標および業務要件とは直接的には紐づかない課題が抽出されることがある。 プロジェクトの目的・目標との関連性を整理し、プロジェクトのスコープに含めるのか否かをお客さまと合意する。 	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> 事前準備をせずにインタビューやアンケートを実施すると、対象者から意見がまったく出ないことや、関係のない業務・システム範囲に意見が拡散しすぎることがある。S1-02-01～S1-02-03 で抽出した課題をもとに、課題の妥当性を担保するための確認事項、妥当性を裏付ける仮説、意見や感想を引き出すための呼び水となる質問、などを準備した上で実施すること。 「S5-01 要件の実施対象決定」で実現対象とするシステム要件の取捨選択を可能とするために、インタビューの記録には誰がどのような課題を表明したか、その課題の重要性や緊急度の認識と理由・根拠を記載しておくこと。 現行システムでシステム化されていない、もしくはシステム仕様が業務とあわないため、運用で対応している事項がないかをステークホルダーにヒアリングすると良い。運用対応を含めた通常業務に慣れてしまうと、課題として認識しにくい側面もあるため。 	
適用技法	
<ul style="list-style-type: none"> インタビュー アンケート ワークショップ ブレインストーミング 	

S1-02-05 課題の原因分析

アクティビティ概要																					
抽出された課題の原因分析を行う。原因分析結果について、ステークホルダーと合意する。																					
インプット		アウトプット																			
● システム課題一覧		● システム課題一覧(原因分析済み)																			
手順																					
<div>1. 原因分析の前に課題の分類、整理を行う。</div> <div>2. 分類、整理を行った課題について、根本原因の分析を行う。</div> <div>3. 原因分析結果について、ステークホルダーへ認識齟齬がないことを確認し、合意する。</div> <div>以下に原因分析の例を挙げる。例えば、下表の下線部のような原因分析が行われる。</div> <table><tr><th>課題 ID</th><th>課題名</th><th>課題内容</th><th>原因</th></tr><tr><td rowspan="2">SP001</td><td rowspan="2">内部統制にあたり各情報にアクセスできるユーザーの権限を適正化</td><td rowspan="2">帳票: 帳票出力後に、各担当者が電子帳票を修正してしまっている。</td><td><u>帳票が抱える原因:</u> <u>編集可能な形式で帳票が出力されている。</u></td></tr><tr><td><u>出力側が抱える原因:</u> <u>通常は商品手配時に、標準販売単価と数量から販売金額を確定するが、一部得意客に対して商品手配後に販売金額調整を行う場合があり、電子帳票を訂正している。</u></td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td rowspan="2">SP005</td><td rowspan="2">現行の運用設計では、障害発生時に復旧目標を満たせない可能性がある。</td><td rowspan="2">現行システムの障害時では、日次で取得しているテープバックアップを利用して再構築することにより復旧することで合意していた。 お客さま運用でバックアップ取得結果の監視を行うことになっているが、運用が徹底されていない。</td><td><u>インフラが抱える原因:</u> <u>サーバがシングル構成になっているため、機器故障時にはバックアップから復旧させるしか方法がない。</u></td></tr><tr><td><u>運用が抱える原因:</u> <u>バックアップ取得結果の監視を行うお客さま担当者が決められていても、異動や退職などのタイミングで引き継ぎが行われていない。</u></td></tr></table>				課題 ID	課題名	課題内容	原因	SP001	内部統制にあたり各情報にアクセスできるユーザーの権限を適正化	帳票: 帳票出力後に、各担当者が電子帳票を修正してしまっている。	<u>帳票が抱える原因:</u> <u>編集可能な形式で帳票が出力されている。</u>	<u>出力側が抱える原因:</u> <u>通常は商品手配時に、標準販売単価と数量から販売金額を確定するが、一部得意客に対して商品手配後に販売金額調整を行う場合があり、電子帳票を訂正している。</u>	SP005	現行の運用設計では、障害発生時に復旧目標を満たせない可能性がある。	現行システムの障害時では、日次で取得しているテープバックアップを利用して再構築することにより復旧することで合意していた。 お客さま運用でバックアップ取得結果の監視を行うことになっているが、運用が徹底されていない。	<u>インフラが抱える原因:</u> <u>サーバがシングル構成になっているため、機器故障時にはバックアップから復旧させるしか方法がない。</u>	<u>運用が抱える原因:</u> <u>バックアップ取得結果の監視を行うお客さま担当者が決められていても、異動や退職などのタイミングで引き継ぎが行われていない。</u>
課題 ID	課題名	課題内容	原因																		
SP001	内部統制にあたり各情報にアクセスできるユーザーの権限を適正化	帳票: 帳票出力後に、各担当者が電子帳票を修正してしまっている。	<u>帳票が抱える原因:</u> <u>編集可能な形式で帳票が出力されている。</u>																		
			<u>出力側が抱える原因:</u> <u>通常は商品手配時に、標準販売単価と数量から販売金額を確定するが、一部得意客に対して商品手配後に販売金額調整を行う場合があり、電子帳票を訂正している。</u>																		
...																		
SP005	現行の運用設計では、障害発生時に復旧目標を満たせない可能性がある。	現行システムの障害時では、日次で取得しているテープバックアップを利用して再構築することにより復旧することで合意していた。 お客さま運用でバックアップ取得結果の監視を行うことになっているが、運用が徹底されていない。	<u>インフラが抱える原因:</u> <u>サーバがシングル構成になっているため、機器故障時にはバックアップから復旧させるしか方法がない。</u>																		
			<u>運用が抱える原因:</u> <u>バックアップ取得結果の監視を行うお客さま担当者が決められていても、異動や退職などのタイミングで引き継ぎが行われていない。</u>																		
【重要】上手く進めるためのポイント、注意事項																					
<div>➤ 課題を分類、整理し、同類課題のグループに対して「なぜ？」を追求することで、根本原因を特定する。 根本的な原因分析ができていない場合、有効な課題解決策を導き出せない。 例えば、「画面が使いにくい」という原因には、「使いやすい画面にする」という単に裏返したような解決策しか定義できない。</div> <div>➤ 解決策ありきで課題と原因を紐づけないこと。 例えば、BI 導入という提案(解決策)を意識しすぎ、無意識のうちに解決策に無理に結びつける原因分析を行ってしまうことがある。</div>																					

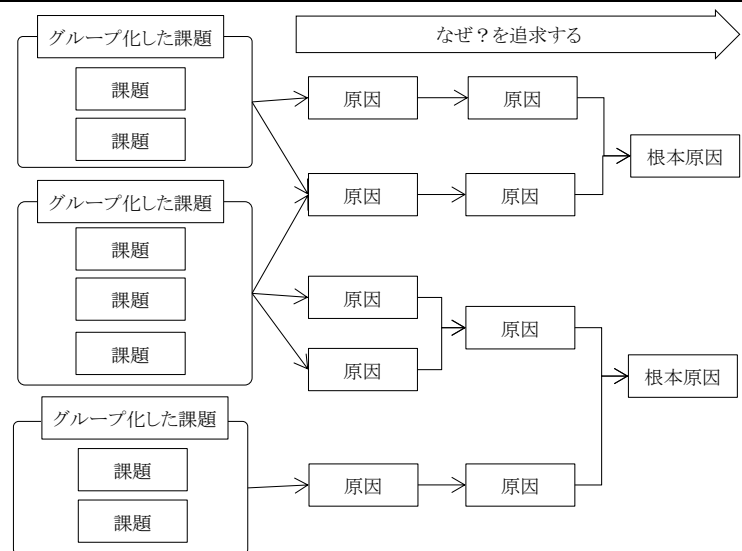


図 1-02-05-1 課題の原因分析イメージ

上手に進めるためのポイント、注意事項

- 課題のグループ化は、同種の事象に集約する程度でよい。
過度にグループ化すると課題が抽象化されてしまい、適切な原因分析ができなくなる。

適用技法

- KI 法
- なぜなぜ分析
- 特性要因分析

S1-03 課題解決の実現手段検討

サブプロセス目的
課題が解決された状態を実現するための手段を明確にする。
サブプロセス概要
課題の根本原因を解消した状態を設定し、その状態を達成する手段を多方向から検討する。その上で実現手段を選定し、詳細化を行う。

S1-03-01 課題解決後の状態設定

アクティビティ概要

適切な実現手段を検討するために、課題解決で実現したい状態を明確にする。特に、課題解決の実現手段が複数想定される場合や、課題の改善・解決に複数の段階が想定される場合は、適切な実現手段を選択するために期待する解決後状態を明確化することが重要である。

解決後の状態について、ステークホルダーへ認識齟齬がないことを確認し、合意する。

以下に課題解決後の状態の設定例を挙げる。例えば、下表の下線部のような解決後の状態の設定が行われる。

課題 ID	課題名	課題内容	改善目標(ゴール)	原因
SP001	内部統制にあたり各情報にアクセスできるユーザーの権限を適正化	帳票: 帳票出力後に、各担当者が電子帳票を修正してしまっている。	<u>編集不可の帳票が出力される顧客毎に設定された販売金額が帳票へ出力されるため、担当者による編集は不要である。</u>	帳票が抱える原因: 編集可能な形式で帳票が出力されている。
				出力側が抱える原因: 通常は商品手配時に、標準販売単価と数量から販売金額を確定するが、一部得意客に対して商品手配後に販売金額調整を行う場合があり、電子帳票を訂正している。
...
SP005	現行の運用設計では、障害発生時に復旧目標を満たせない可能性がある。	現行システムの障害時には、日次で取得しているテープバックアップを利用して再構築することにより復旧することで合意していた。 お客さま運用でバックアップ取得結果の監視を行うことになっているが、運用が徹底されていない。	<u>複数の復旧方法により、障害発生時点の1営業日前の状態への復旧が担保されている。</u>	インフラが抱える原因: サーバがシングル構成になっているため、機器故障時にはバックアップから復旧させるしか方法がない。
				運用が抱える原因: バックアップ取得結果の監視を行うお客さま担当者が決められていても、異動や退職などのタイミングで引き継ぎが行われていない。

インプット		アウトプット	
<ul style="list-style-type: none">システム課題一覧(原因分析済み)		<ul style="list-style-type: none">課題解決後の状態設定資料システム課題一覧(課題解決後の状態設定済み)	
手順			
—			
【重要】上手く進めるためのポイント、注意事項			
<p>➤ 課題解決後の状態は、解決を達成したか否かを評価できる基準や、計測できる効果を設定する。</p>			
上手く進めるためのポイント、注意事項			
—			
適用技法			
<ul style="list-style-type: none">(特になし)			

S1-03-02 実現手段の方向性検討

アクティビティ概要				
<p>課題の根本原因を解消するための詳細な手段を検討する前に、採り得る解決手段を多角的に検討し、方向性を検討する。</p> <p>実現手段の方向性について、ステークホルダーへ認識齟齬がないことを確認し、合意する。</p> <p>以下に実現手段の方向性の検討例を挙げる。例えば、下表の下線部のような実現手段の方向性検討が行われる。</p>				
課題 ID	課題名	課題内容	改善目標(ゴール)	解決手段(実現手段)
SP001	内部統制に当たり各情報にアクセスできるユーザーの権限を適正化	帳票:帳票出力後に、各担当者が電子帳票を修正してしまっている。	編集不可の帳票が出力される顧客毎に設定された販売金額が帳票へ出力されるため、担当者による編集は不要である。	<u>内部統制上の対応が必要な帳票について</u> ①項目を編集不可能とする ②内部統制上、編集が許されない項目のみを編集不可能とする
...
SP005	現行の運用設計では、障害発生時に復旧目標を満たせない可能性がある。	現行システムの障害時では、日次で取得しているテープバックアップを利用して再構築することにより復旧することで合意していた。 お客さま運用でバックアップ取得結果の監視を行うことになっているが、運用が徹底されていない。	障害発生時点の1営業日前の状態への復旧が担保されている。	<u>バックアップ取得について</u> ①保守ベンダーが手動で監視する ②自動で監視する <u>復旧の方法について</u> ①バックアップメディアから復旧する ②待機系サーバを構築し、そこからの切替で復旧する
インプット			アウトプット	
<ul style="list-style-type: none"> 課題解決後の状態設定資料 システム課題一覧(課題解決後の状態設定済み) 			<ul style="list-style-type: none"> 実現手段方向性検討資料 	
手順				
—				
【重要】上手く進めるためのポイント、注意事項				
—				
上手く進めるためのポイント、注意事項				
<p>➤ 方向性を検討する際は、検討の観点が網羅的になるように留意すること。</p> <p>例えば、レスポンスを改善するためには、アプリケーション、ミドルウェア、ハードウェア、ネットワークの各層でどのような解決の実現手段があるのかを全体として検討することにより、多様な解決策から適切なものを選択できる。</p>				
適用技法				
<ul style="list-style-type: none"> KJ 法 				

S1-03-03 実現手段の詳細化

アクティビティ概要	
「S1-03-02 実現手段の方向性検討」で検討した方向性に沿って、課題を解決する為の実現手段の検討を行う。 その際、詳細化した実現手段が、解決後の状態を達成するものであるかの整合性をステークホルダーと確認し、合意する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 実現手段方向性検討資料 ● システム課題一覧(課題解決後の状態設定済み) 	<ul style="list-style-type: none"> ● システム課題一覧(ステークホルダー確認済み)
手順	
1. 実現手段の方向性に基づき、具体的な実現手段の案を検討する。 2. 方式設計が必要な実現手段の場合は、実現手段の案と実現手段方向性検討資料を方式設計担当者へ連携する。 3. 詳細化した実現手段が、解決後の状態を達成するものであるかの整合性を確認する。 4. 詳細化した実現手段について、ステークホルダーに認識齟齬がないことを確認し、合意する。	
【重要】上手く進めるためのポイント、注意事項	
➤ 実現手段の詳細化を行う際に、細部のみ検討が集中してしまうと、本来の目的から逸れてしまうことがある。 課題解決後状態への効果を常に意識して実現手段を詳細化する。	
上手く進めるためのポイント、注意事項	
➤ 実現手段が、単に課題の裏返しの表現となっており、具体性がない場合がある。 課題の原因分析が十分でないことが考えられる為、「S1-02 課題の抽出と原因分析」に立ち返って再度原因分析を行う。 ➤ 一つの根本原因に対し、複数の実現手段を検討する。課題解決後の状態による検証に基づき、各実現手段の比較評価を行う。 ➤ 解決手段は必ずしもシステム化とは限らない。 例えば、S1-03-02 の表で例示した課題 ID「SP005」の解決手段は、日次バックアップが確実に行われたことを担保する体制を運用レベルで作れば、待機系サーバを作るというシステム的手段を使わなくても問題は解決する。	
<div style="display: flex; justify-content: space-around; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px;">課題</div> <div style="font-size: 24px;">→</div> <div style="border: 1px solid black; padding: 5px;">実現手段</div> <div style="margin-left: 10px;">×課題の裏返しの実現手段</div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px;">課題</div> <div style="font-size: 24px;">→</div> <div style="border: 1px solid black; padding: 5px;">原因</div> <div style="font-size: 24px;">→</div> <div style="border: 1px solid black; padding: 5px;">実現手段</div> <div style="margin-left: 10px;">×不十分な原因分析</div> </div>	
<div style="text-align: center; margin-top: 20px;"> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>■原因分析の追求</p> <div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">課題</div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 60px;">原因</div> <div style="border: 1px solid black; padding: 5px; width: 60px;">原因</div> </div> <div style="border: 1px solid black; padding: 5px; width: 80px; margin-top: 10px; text-align: center;">根本原因</div> </div> <div style="text-align: center;"> <p>■複数の実現手段の検討</p> <div style="display: flex; justify-content: center; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 60px;">実現手段</div> <div style="border: 1px solid black; padding: 5px; width: 60px;">実現手段</div> <div style="border: 1px solid black; padding: 5px; width: 60px;">実現手段</div> </div> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="border-top: 1px dashed black; width: 40%;"></div> <div style="text-align: center;"> <p>■課題解決後の状態による検証</p> <div style="border-top: 1px dashed black; width: 40%;"></div> </div> </div> </div>	
<p>図 1-03-03-1 実現手段の詳細化のイメージ</p>	
適用技法	
<ul style="list-style-type: none"> ● （特になし） 	

S1-04 機能の整理

サブプロセス目的
「S1-03 課題解決の実現手段検討」で詳細化された実現手段をもとに、どのようなシステム機能を実現する必要があるのかを明らかにする。
サブプロセス概要
「S1-03 課題解決の実現手段検討」で詳細化された実現手段をもとに、どのようなシステム機能を実現する必要があるのかを整理し、システム全体の整合性・一貫性が担保されたシステム機能として整理されていることを確認する。 その上で、ステークホルダーに認識齟齬がないことを確認する。

S1-04-01 システム要求のリストアップ

アクティビティ概要	
「S1-03 課題解決の実現手段検討」で抽出・具体化された実現手段をシステム要求としてシステム要求一覧にまとめる。 業務要件定義において抽出されたシステム要求(業務課題から付随的に抽出されたシステム要求)と合わせて管理できるようにする。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム課題一覧(ステークホルダー確認済み) ● システム要求一覧(業務要件定義時) 	<ul style="list-style-type: none"> ● システム要求一覧
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S1-04-02 機能の整理

アクティビティ概要	
システム機能の振る舞いとシステム間の関連や境界を明確にする。	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 システム機能一覧(業務要件定義時) 	<ul style="list-style-type: none"> システム機能一覧(整理済み) システム機能俯瞰図 システムフロー
手順	
<ol style="list-style-type: none"> 「S1-04-01 システム要求のリストアップ」でリストアップしたシステム要求をもとに、実現するシステム機能をシステム機能一覧に定義する。 システム機能を、画面・帳票・バッチ・外部IF等の処理方式で分類する。 システム機能一覧をもとに、サブシステムを定義する。 システム機能一覧で整理した機能を、定義したサブシステムへ分類する。 サブシステムの分類で、業務機能と直接対応しないサブシステム(内部共通機能など)を新たに定義した場合は、システム機能一覧を修正する。 サブシステム間で受け渡す情報、および外部システム間で受け渡す情報を整理し、システム機能俯瞰図を作成する。 業務の流れの視点で、DB エンティティ等を介した各システム機能のつながりを整理し、システムフローを作成する。 フローがつかない場合は、必要なシステム機能を分析し、システム機能の追加を検討する。 	
【重要】上手く進めるためのポイント、注意事項	
<p>システム機能一覧</p> <hr/> <ul style="list-style-type: none"> システム要求一覧とシステム機能一覧との突合せを行う。 システム要求一覧にリストアップした要求を軸に、システム機能一覧を確認することで、要求に対する機能の漏れを確認できる。 <p>システム機能俯瞰図</p> <hr/> <ul style="list-style-type: none"> システム機能俯瞰図を作成する際には、接続先外部システムやそれらとの外部IFの抽出漏れに注意する。 外部システムや外部IFの抽出漏れは、インフラ設計、方式設計、ソフトウェア設計と影響範囲が広く、コスト影響も大きい。 システム機能一覧に挙がっている外部IFを集約してシステム機能俯瞰図に反映するだけでなく、現行システムの接続先システム一覧や外部IF一覧との対比や、お客さま企業内外のシステムの責務や保有データの観点から自システムとのIF 要否を確認する、などの手段で接続先外部システム、外部IFの抽出漏れを防止する。 	
上手く進めるためのポイント、注意事項	
<p>システム機能一覧</p> <hr/> <ul style="list-style-type: none"> システム機能一覧の処理方式は、基本的には画面とバッチに分類されるが、それらに含まれる帳票と外部IFを個別に機能として抽出することで、「S2 機能要件の定義」の定義対象を明確にすることができる。 <p>システム機能俯瞰図</p> <hr/> <ul style="list-style-type: none"> 手動でのデータ取り込みや、外部システムで出力した紙媒体を参照した手入力でのデータ連携が抜けがちなので注意する。 サブシステム間のデータ受け渡しが少なく疎結合なサブシステム分割を行う。 サブシステムごとにDB エンティティを持ってオーナーとなり、オーナーエンティティのCRUDの主体となるシステム機能をサブシステムに集約することが基本的なサブシステム分割の考え方である。これによって、サブシステム内にはオーナーエンティティを主アクセス対象とするシステム機能が集約され、サブシステム間が疎結合の良いサブシステム分割ができる。不適切なサブシステム分割をした場合、サブシステム間IFやサブシステムをまたぐDB アクセスが増加・複雑化、サブシステムを担当する組織、チーム間のコミュニケーションの増加・複雑化し、開発生産性を低下させる要因になる。DB エンティティはお客さま業務の集合ごとにおおよそ分類されるため、サブシステムは一定粒度のお客さま業務の集合単位と同じになる。目安として、業務階層定義上のレベル1～2の業務ごとにサブシステムを定義するのが適当である。 <p>システムフロー</p> <hr/> <ul style="list-style-type: none"> 作成単位は、業務階層定義の業務階層を元に検討する。 	

業務階層定義のレベル3単位にシステム処理フローを定義するなど。

- 以下の点を意識して作成する。
 - ✓ スイムレーンには業務フローに定義されたアクター名、構築対象のシステム名、関連する外部システム名を定義して、それぞれの関係が表現できるようにすること。
 - ✓ 構築対象システムの機能は、UIを持つ機能、UIを持たない機能(バッチ機能、外部F機能など)の違いが、分かるようなアイコンにする。視覚的に理解しやすいようにする。
 - ✓ 外部システム側の外部F機能名も定義する。外部Fは、他社システムからのシステム要求により発生することもある為、定義が漏れやすい。外部システム側の取り込み機能も必要になることを留意すること。
 - ✓ 機能だけでなく、関連するエンティティと連携データも定義する。
 - ✓ UIシステム機能を利用した業務作業の流れと、システム機能が利用するデータの流れ(エンティティとの関係)を記述する。業務の流れとデータの流れは、意味が異なる為、実線と点線などの使い分けをすること。

適用技法

- (特になし)

S1-04-03 機能間の整合性、一貫性の確認

アクティビティ概要	
システム機能が過不足なく整合性、一貫性の取れた状態で抽出できていることを確認する。 漏れや誤り等が判明した場合は、「S1-02 課題の抽出と原因分析」プロセスに戻る。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム機能一覧(整理済み) ● システム機能俯瞰図 ● システムフロー 	<ul style="list-style-type: none"> ● システム機能一覧(整合性、一貫性確認済み) ● システム機能俯瞰図(整合性、一貫性確認済み) ● システムフロー(整合性、一貫性確認済み)
手順	
—	
【重要】上手に進めるためのポイント、注意事項	
—	
上手に進めるためのポイント、注意事項	
システム機能一覧内で確認する整合性、一貫性	
<ul style="list-style-type: none"> ➤ 同一機能分類内の機能構成が、データライフサイクルを網羅しているか確認する。 例えば、L2層「〇〇管理機能」の配下L3層に「〇〇登録機能」「〇〇照会機能」「〇〇変更機能」「〇〇削除機能」のように、作成、参照、更新、削除の CRUD に対応した機能が揃っていることを確認する。 ➤ 異なる機能分類間で機能を比較する。類似した機能を比較して、機能の定義に漏れがないかを確認する。 例えば、L2層「顧客管理機能」にはL3層機能「顧客一覧帳票」が定義されているが、別のL2層「仕入先管理機能」には「仕入先一覧帳票」が未定義の場合は、機能の定義に漏れがある可能性がある。 	
システム機能一覧と他の成果物を比較して確認する整合性、一貫性	
<ul style="list-style-type: none"> ➤ システム機能俯瞰図に定義された機能と、システム機能一覧に定義された機能を比較して、定義された機能に差異がないことを確認する。 システム機能俯瞰図の機能の粒度が大きく括られているなど、システム機能一覧とシステム機能俯瞰図の機能粒度が異なっている場合は、機能一覧の上位階層の機能分類名と突き合わせて確認を行う。 ➤ システムフローに定義された機能が漏れなく、システム機能一覧に定義されていることを確認する。 	
システムフローとシステム機能俯瞰図を比較して確認する整合性、一貫性	
<ul style="list-style-type: none"> ➤ システムフローに記載されたシステム機能間の関連から、サブシステムを跨るシステム機能間の関連を抽出し、その関係が漏れなく、システム機能俯瞰図に表現されていることを確認する。 ➤ システムフローに記載された外部システムやシステム利用者との関係が漏れなく、システム機能俯瞰図に表現されていることを確認する。 	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S1-04-04 機能のステークホルダー確認

アクティビティ概要	
整理したシステム機能について、システム機能の抜け漏れや重複がなく、システム要求を正しく反映した適当な機能要求内容であることを、ステークホルダーと確認する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム機能一覧(整合性、一貫性確認済み) ● システム機能俯瞰図(整合性、一貫性確認済み) ● システムフロー(整合性、一貫性確認済み) ● システム要求一覧 	<ul style="list-style-type: none"> ● システム機能一覧(ステークホルダー確認済み) ● システム機能俯瞰図(ステークホルダー確認済み) ● システムフロー(ステークホルダー確認済み)
手順	
<ol style="list-style-type: none"> 1. システム要求とシステム機能の対応関係、個々のシステム機能内容、システムの流れをステークホルダーに説明する。 2. ステークホルダーは、システム要求を実現するシステム機能に抜け漏れなく、システム要求を正しく反映した適切なシステム機能内容であることを確認する。 	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 他システムとの関連がある場合は、IF 方式や IF レイアウト等を当該システムの担当者とすり合わせる。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ 業務フローと、各説明資料を対応付けて説明すると、業務に対するシステム化の観点から機能の過不足を確認しやすくなる。</p> <p>➤ 機能を確認する際は、一度に全ての範囲を説明しようとせず、業務要件と紐づく単位などを個別に切り出して説明すると、ステークホルダーが理解しやすい。</p> <p>➤ 後続の開発作業をサブシステム構成に合わせて分割して行う場合は、サブシステム分割の考え方とサブシステム構成をお客さまに理解頂き、後続作業の分割単位と範囲の認識を合わせる。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S2 機能要件の定義

S2-01 機能要件の定義

サブプロセス目的
システム機能要件を、画面機能要件・帳票機能要件・外部 IF 機能要件・バッチ機能要件および論理データモデルとして詳細化し、設計工程以降の工数見積もりと、開発スコープを明確にする。
サブプロセス概要
画面、帳票、外部 IF、バッチの機能ごとに、各機能の個別要件を明確にする。機能で取り扱うデータは、論理データモデルで明確にする。方式設計で定めた、入出力共通仕様、画面標準、帳票標準、連携標準、命名規約、処理パターンなどが、要件を満たせない場合、方式設計担当者へフィードバックし、方式設計との調整を行う。機能要件定義では原則、お客さまに実現を約束することを明確化し、それを実現する方法や設計は設計工程で具体化する、という考え方で定義事項の妥当性を判断する。(計画時点でお客さまと調整・合意する)

S2-01-01 画面機能の要件定義

アクティビティ概要	
画面機能が提供する画面を一覧化し、各画面間の遷移を明確化する。また、各画面の入出力項目や項目レイアウト等の画面機能要件を明確化する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 ● システム機能一覧(ステークホルダー確認済み) ● 画面設計(As-Is) ● 概念データモデル(To-Be) 	<ul style="list-style-type: none"> ● 画面機能要件定義
手順	
<ol style="list-style-type: none"> 1. 画面一覧の定義 画面機能を個々の画面に細分化し、画面一覧にまとめる。 2. 画面遷移、入出力項目と配置、イベントの定義 業務遂行に必要な画面の遷移を明確にする。 また画面入出力項目とレイアウト、その属性や制約事項、及びボタン等を操作した際のシステムの振る舞いを明確にする。 3. 定義内容のお客さまへの確認、合意 	
【重要】上手く進めるためのポイント、注意事項	
画面レイアウト <ul style="list-style-type: none"> ➤ モックアップツールや HTML で作成するとより実現性が高いレイアウトとなる。 ➤ 画面レイアウトパターン、画面遷移パターン、画面サイズ、配色、タブオーダー、等を画面設計標準として定義された、全画面の統一性確保や機能部品の共通利用による生産性向上などを考慮した機能要件定義を行うこと。(通常、画面設計標準は方式設計担当者が行うため、要件定義フレームワークの解説外としています) ➤ 利用者権限、出力対象データのステータス、等に応じた特殊制御要求がないか確認する。(画面遷移、入出力項目、イベントも同様) 	
入出力項目 <ul style="list-style-type: none"> ➤ 情報が出力される理由や出力される情報の意味を明確にする。 出力される情報は機能の目的に整合しているか、無駄や余分な情報が含まれていないかを確認する。 	
上手く進めるためのポイント、注意事項	
画面遷移 <ul style="list-style-type: none"> ➤ 上から下、左から右に遷移するように画面を配置すると、遷移を直観的に理解しやすくなる。[1] ➤ 正常系の遷移と、異常系の遷移の判別がしやすいように、一目で区別できるような書き分け方をする。[1] ➤ システム機能単位に、画面を整理すると、抜け漏れや重複を防ぐことができ、全体を把握しやすくなる。 	
画面レイアウト <ul style="list-style-type: none"> ➤ ユーザーの操作性、開発者の開発生産性やテスト容易性を低下させるような、複雑な画面レイアウトは極力避けるようお客さまと調整する。 	
入出力項目 <ul style="list-style-type: none"> ➤ 概念データモデル(To-Be)および画面設計(As-Is)より入出力項目を検討する。 ➤ 複数画面で入力を行う機能の場合は、各入力項目の配置先画面の妥当性を業務の流れの観点から確認する。 	

画面イベント

- 条件分岐を表現する際には、ディシジョンテーブル(複数の条件についてその成立／不成立の組み合わせを表形式に展開し、それぞれの場合に対応する結果を記述する)等を利用して整理すると漏れをチェックしやすくなる。
- ロジックには、業務ルールに基づいた変更できないもの(外部要因)と、システムの構造・方式・技術等の制約から生まれた変更できるもの(内部要因)がある。システム要件定義では外部要因に依存するロジックを明確にする。
(原則、内部要因に依存するロジックは”要件”としてお客さま合意を得る必要はなく、設計工程で検討すればよい)

適用技法

- (特になし)

S2-01-02 帳票機能の要件定義

アクティビティ概要	
帳票機能が提供する帳票を一覧化し、各帳票の出力項目や項目レイアウト、編集仕様等の帳票機能要件を明確化する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 ● システム機能一覧(ステークホルダー確認済み) ● 帳票設計(As-Is) ● 概念データモデル(To-Be) 	<ul style="list-style-type: none"> ● 帳票機能要件定義
手順	
<ol style="list-style-type: none"> 1. 帳票一覧の定義 システムが出力する帳票を帳票一覧にまとめる。 2. 帳票の出力項目と配置の定義 帳票出力項目や項目レイアウト、編集仕様等を明確にする。 3. 定義内容のお客さまへの確認、合意 	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
帳票レイアウト	
<ul style="list-style-type: none"> ➤ データ量によって印字する行数が増減する場合、改ページ条件を考慮した帳票レイアウトを作成する。[1] ➤ 印刷後に、穴をあけてファイリングする帳票は、穴開け箇所の余白設定が必要な為、確認しておくこと。 ➤ 実業務で紙の伝票が用いられている場合は、紙伝票と比較することで不足項目のチェックをしやすくなる。 ➤ システムが印刷内容を動的に変える箇所と、固定印刷箇所の色付けを分ける等で、印刷項目の過不足チェックがしやすくなる。 	
帳票項目、編集仕様定義	
<ul style="list-style-type: none"> ➤ 概念データモデル(To-Be)および帳票設計(As-Is)より入出力項目を検討する。 ➤ 利用者権限や出力対象データのステータス、出力タイミング等に応じた編集仕様や出力項目制御などの要求がないか注意する。 ➤ 帳票レイアウトと帳票項目説明間で項目ごとに同番号を付与しておくなど、項目突合せをしやすくする。 ➤ 帳票の再出力可否、出力制限、出力中止、等を考慮した検討事項を漏らしやすいので注意する。 ➤ 帳票上の固定出力項目とプロジェクトの用語集の整合性、帳票設計標準への準拠など、必要なルールへの適合を確認する。 	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S2-01-03 外部 IF 機能の要件定義

アクティビティ概要		
接続先外部システムとの IF 機能を一覧化し、接続先外部システム、接続方式、入出力種別等の IF 機能要件を明確化する。		
インプット		アウトプット
<ul style="list-style-type: none"> システム要求一覧 システム機能一覧(ステークホルダー確認済み) 外部 IF 設計(As-Is) 概念データモデル(To-Be) 		<ul style="list-style-type: none"> 外部 IF 機能要件定義
手順		
<ol style="list-style-type: none"> 外部システム関連図の作成 接続先外部システムとのデータの送受信関係を IF 単位で俯瞰的に図示する。 外部 IF 一覧の定義 接続先外部システムとの間でデータをやりとりする IF を一覧形式にまとめる。 外部システムとの IF を実現するために使用する接続方式を明確にする。 外部 IF のデータ項目の定義 外部 IF に含まれるデータ項目と 定義内容のお客さまへの確認、合意 		
【重要】上手く進めるためのポイント、注意事項		
全体		
<ul style="list-style-type: none"> 外部システムとの連携は、主システムがどちらかによって進め方が変わってくる場合がある。 例えば、基幹システムやパッケージが既に存在し、今回構築するシステムが周辺システムである場合、連携仕様は基幹システムやパッケージの仕様に合わせる必要がある。その場合は、相手先に連携仕様を提示してもらう必要がある。 外部接続先システム、外部 IF の抽出漏れは、アプリケーション・ソフトウェア方式・インフラなど広範囲に、コスト・スケジュール面の大きな影響を及ぼす可能性がある。ここまでに作業も含め、外部 IF 要件の抽出には慎重を期すこと。 		
外部 IF 項目		
<ul style="list-style-type: none"> 既存の外部 IF は現行仕様を踏襲するが多いが、「現行通り」という要求が指す内容を正確に把握する。 実データやレイアウト定義ファイルを入手し、設計書と合わせて確認すると良い。 (項目が未設定時は空白なのか、NULL なのか、設定可能な場合、どのような範囲の値が入力可能なのか、など) 		
上手く進めるためのポイント、注意事項		
全体		
<ul style="list-style-type: none"> 遵守すべきセキュリティポリシーが存在しないかを確認する。 		
外部 IF 一覧		
<ul style="list-style-type: none"> 以下資料との突き合わせを行い、外部 IF の定義漏れを防止する。 		
No	資料	確認内容
1	システムフロー	システムフローに現れる外部 IF が、外部 IF 一覧に漏れなく定義されていることを確認する。
2	外部 IF 一覧(As-Is)	現行システムと次期システムの外部 IF 一覧を突き合わせ、必要な現行 IF が踏襲されていることを確認する。
<ul style="list-style-type: none"> 接続先外部システムまたは自システムに依存する、送受信タイミングの制約がある場合は、具体的な時刻や制約理由を示すなど明確に定義し、関係者と認識を合わせる。 選択する接続処理方式について、エラー発生時のリトライ可否を確認し、リトライ回数/リトライ間隔/ログ出力方式などのリトライ要件を明確にすること。 [1] 		

- 廃止または変更する、現行システム外部 IF がある場合、新システムおよび接続先外部システムにおける影響を確認する。
- 相手のシステムが既存システムか新規システムかを明記する。[1]
相手のシステムが既存ならば外部 IF 要件が変更されることは少ないが、新規システムの場合、外部 IF 要件が変更される可能性がある。

外部システム関連図

- 外部システム関連図の作成後、外部 IF 一覧の照合を行うと漏れを発見できる。

外部 IF 項目

- 外部 IF が複数種類のレコードから構成される場合には、外部 IF 項目定義をレコード種別ごと(例えば、ヘッダー部、明細部、トレーラー部など)に定義し、各レコード種別の並び方を定義する。[1]

適用技法

- (特になし)

S2-01-04 バッチ機能の要件定義

アクティビティ概要	
バッチ機能を一覧化し、処理概要やインプット、アウトプット、処理サイクル等のバッチ機能要件を明確化する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 ● システム機能一覧(ステークホルダー確認済み) ● バッチ設計(As-Is) ● 概念データモデル(To-Be) 	<ul style="list-style-type: none"> ● バッチ機能要件定義
手順	
<ol style="list-style-type: none"> 1. バッチ一覧の定義 バッチ機能を一覧化する。 2. バッチ機能要件の定義 バッチ処理の入力項目として利用する入力元のデータ項目と、バッチ処理の出力項目を明確にする。出力項目の算出方法や条件を明確にする。 3. 定義内容のお客さまへの確認、合意 	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
バッチ機能一覧 <hr/> <ul style="list-style-type: none"> ➤ バッチの処理サイクル、処理タイミング、起動の条件により、起動方式(スケジュール起動、先行ジョブ終了時起動、オンラインバッチ起動、オペレータ起動など)を整理する。リソース見積や運用設計など非機能要件、コストへの影響が想定される。 	
バッチ処理定義 <hr/> <ul style="list-style-type: none"> ➤ 入力から出力への想定データ変化と、バッチ処理概要の整合を確認する。 ➤ 業務的な理由から、開始時間/終了時間などに制約がかかる場合がある。 業務時間による実行時間の要件、休業日中の処理などカレンダーによる要件、月次の処理量の増加等を検討し、要件を満たすよう定義する。 	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S2-01-05 論理データモデルの定義

アクティビティ概要	
各機能要件で定義した機能が利用するエンティティと項目、エンティティ間のルーション等を明確にする。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 ● システム機能一覧(ステークホルダー確認済み) ● データモデル(As-Is) ● 画面機能要件定義 ● 帳票機能要件定義 ● 外部IF 機能要件定義 ● バッチ機能要件定義 ● 概念データモデル(To-Be) 	<ul style="list-style-type: none"> ● 論理データモデル定義 (ER図、エンティティ一覧、ドメイン定義、エンティティ定義、コード定義) ● CRUD 図
手順	
<ol style="list-style-type: none"> 1. ER 図を作成し、属性項目を含むエンティティ、エンティティ間の相関関係を可視化する。 2. ER 図に定義したエンティティを一覧化する。 3. 定義域(データ型、データの取り得る値・範囲など)が同じ属性項目をグループ化し、ドメインとして定義する。 例えば、「自宅電話番号」と「勤務先電話番号」という属性項目から「電話番号」ドメインを抽出する。 4. エンティティ属性項目のキー項目、型、桁、等の詳細を定義する。 5. 属性項目が特定の値のバリエーションを持つ場合や、番号の採番方式などの特定のルールを持つ場合には、コードとして定義する。 6. エンティティと、各エンティティデータの作成(C)、参照(R)、更新(U)、削除(D)を行う機能の関係を CRUD 図に整理する。 エンティティごとに C/R/U/D を行う機能数を集計し、エンティティデータのライフサイクルの観点から、機能の充足性を確認する。 7. 定義内容をお客さまと確認し、合意する。 	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
ER 図 <ul style="list-style-type: none"> ➤ エンティティをイベント／リソース／サマリの分類でタイプ分けして、色、網掛け、枠囲みで表現すると、概観を掴みやすくなる。[1] <ul style="list-style-type: none"> ✓ イベント : 業務活動によって発生・増加する情報を管理するエンティティ(受注、発注など) ✓ リソース : 実際に存在するモノを管理するエンティティ(顧客マスタ、商品マスタなど) ✓ サマリ : 業務活動によって発生した情報の集計結果を管理するエンティティ(売上集計など) ➤ ER 図中のエンティティにデータ作成、および更新の時期をコメントとして追記すると、業務とエンティティのおおよその関係が把握可能となり、理解しやすくなる。 ➤ イベントのエンティティをデータの発生順に並べて ER 図に表現すると、業務の流れにそった確認ができ、理解しやすい。 	
エンティティ一覧 <ul style="list-style-type: none"> ➤ 現行システムが存在し、かつ対象業務も大きな変更なく新システムで踏襲する場合は、新システムは現行のデータモデルを踏襲することが多い。新システムと現行システムのエンティティ対応関係を整理することで、新システムにおけるデータモデル変更要求の対応結果確認が容易になる。例えば新システムでは、「性能要求実現」、「新アーキテクチャ採用」等のためにデータモデル設計方針を変更することが考えられる。 ➤ データ件数や保存期間は、データモデルを決める要素となる。その理由となる規則や法令などを含め、エンティティ一覧に記載すると良い。 	
CRUD 図 <ul style="list-style-type: none"> ➤ CRUD 図を作成し、各機能要件から抽出されたシステム機能とエンティティの関係をエンティティデータのライフサイクルの観点で整理することで、不足している機能や重複、タイミングの齟齬等を検証する。 	

- CRUD 図のエンティティをイベント系エンティティとリソース系エンティティに分離して記述すると、システム機能の漏れを発見しやすい。イベント系エンティティとリソース系エンティティを分離し、業務活動によって発生・増加するイベント系エンティティに注目しやすくすることで、業務に対するシステム機能の漏れが、発見しやすくなる。

お客さま確認

- 業務やシステムの規模が大きく、論理データモデル全体が複雑化する場合は、業務階層定義のL1やL2の業務単位に分けて、お客さまに説明するとよい。
- 業務要件定義で作成した概念データモデル定義と比較することで、論理設計で詳細化した箇所・内容を確認しやすい。
- 画面要件定義や帳票要件定義を参照しながら、お客さまにエンティティの属性項目を確認頂くと、妥当性を判断しやすい。
- 複雑なデータ構造を確認するには、論理データモデルに実データを当てはめて補足するとよい。

適用技法

- Peter Chen 記法
- IDEF1X (Integration Definition)記法
- IE (Information Engineering)記法

S3 非機能要件の定義

S3-01 モデルシステムの選定

サブプロセス目的		
開発するシステムイメージに近いモデルシステムを選定し、そのモデルシステムに定義された非機能要求項目のマトリクスのベース値を参考に非機能要求を検討することで、効率的に非機能要件を定義できるようにする。		
サブプロセス概要		
<p>IPA/SEC の非機能要求グレードは、非機能要求を以下の6つの大項目に整理しており、さらに非機能要求を 236 項目のマトリクスに展開している。マトリクスには、0～5の6段階の対応レベルが定義されており、非機能要求の対応レベル決めに利用できる。</p> <p>(注)本フレームワークでは以下の非機能要求グレードに「ネットワーク要件の定義」「テスト要件の定義」を取り込んでいる。</p>		
大項目	要求の例	確認結果に基づき、実施する対策例
可用性	<ul style="list-style-type: none"> ✓ 運用スケジュール(稼働時間・停止予定など) ✓ 障害、災害時における稼働目標 	<ul style="list-style-type: none"> ✓ 機器の冗長化やバックアップセンターの設置 ✓ 復旧・回復方法及び体制の確立
性能・拡張性	<ul style="list-style-type: none"> ✓ 業務量および今後の増加見込み ✓ システム化対象業務の特性(ピーク時、通常時、縮退時など) 	<ul style="list-style-type: none"> ✓ 性能目標値を意識したサイジング ✓ 将来へ向けた機器・ネットワークなどのサイズと配置＝キャパシティ・プランニング
運用・保守性	<ul style="list-style-type: none"> ✓ 運用中に求められるシステムの稼働レベル ✓ 問題発生時の対応レベル 	<ul style="list-style-type: none"> ✓ 監視手段およびバックアップ方式の確立 ✓ 問題発生時の役割分担、体制、訓練、マニュアルの整備
移行性	<ul style="list-style-type: none"> ✓ 新システムへの移行期間および移行方法 ✓ 移行対象資産の種類および移行量 	<ul style="list-style-type: none"> ✓ 移行スケジュール立案、移行ツール開発 ✓ 移行体制の確立、移行リハーサルの実施
セキュリティ	<ul style="list-style-type: none"> ✓ 利用制限 ✓ 不正アクセスの防止 	<ul style="list-style-type: none"> ✓ アクセス制限、データの秘匿 ✓ 不正の追跡、監視、検知
システム環境・エコロジー	<ul style="list-style-type: none"> ✓ 耐震／免震、重量／空間、温度／湿度、騒音など、システム環境に関する事項 ✓ CO₂ 排出量や消費エネルギーなど、エコロジーに関する事項 	<ul style="list-style-type: none"> ✓ 規格や電気設備にあった機器の選別 ✓ 環境負荷を低減させる構成

表 3-01-1 非機能要求グレードの6大項目[2]

また、非機能要求グレードでは、以下の3つのモデルシステムとその特徴が定義されており、そのモデルシステム毎に非機能要求項目のマトリクスのベース値(初期値)が設定されている。このベース値を参考に、非機能要求の対応レベルを決めることができるようになっている。

- ✓ 社会的影響が殆ど無いシステム
- ✓ 社会的影響が限定されているシステム
- ✓ 社会的影響が極めて大きいシステム

本サブプロセスでは、この非機能要求グレードで定義された3つのモデルシステムを参考に、開発するシステムのイメージ(社会的影響度合いなど)にもっとも近いモデルシステムを選定する。

S3-01-01 モデルシステムの選定

アクティビティ概要	
<p>非機能要求グレードで定義された3つのモデルシステムを参考に、開発するシステムのイメージ(社会的影響度合いなど)にもっとも近いモデルシステムを選定する。S3-02以降のプロセスでは、本アクティビティで選定したモデルシステムに示されるベース値を参考に、非機能要求の対応レベルを調整・設定し、その上で非機能要件を具体化する。</p> <p>ただし、必ずしも本アクティビティから始める必要はない。大まかな非機能要求しか決まっていない状態では、「モデルシステムの選定」から開始するのが良いが、ある程度非機能要求の項目が明確になっている状態では、本アクティビティをスキップして個々の非機能要求の対応レベルを決定していくという進め方でも良い。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> 非機能要求グレード(モデルシステムシート) 	<ul style="list-style-type: none"> (特になし)
手順	
<ol style="list-style-type: none"> 開発するシステムのイメージ(システムの役割やシステム機能が低下または利用不可能な状態に陥った場合の影響の程度など)にもっとも近いモデルシステムを、非機能要求グレードのモデルシステムシートで説明されている「モデルシステムの概要」を参照して選択する。 モデルシステムシートで説明されている選択したモデルシステムの「非機能要求の概要」と開発するシステムの非機能要求の概要を比較し、モデルシステム選択の妥当性を確認する。 	
【重要】上手に進めるためのポイント、注意事項	
<p>※モデルシステムの具体的な内容については、非機能要求グレードの「システム基盤の非機能要求に関するグレード表」を参照</p> <p>※非機能要求グレードの具体的な利用方法については、非機能要求グレードの「非機能要求グレード利用ガイド」を参照</p> <p>「S3-02」以降の非機能要件定義アクティビティに関する共通ポイント</p> <hr/> <p>➤ 「S3-02」以降のアクティビティで定義されている非機能要求の項目全てを定義する必要はない。プロジェクトの特性などに合わせ、検討する非機能要求項目を選定して良い。お客さま要求事項がない非機能要求項目は“要求なし”とすればよい。お客さまからの要求事項が存在しないことを明示することは重要。</p> <p>ただし、考慮不要と判断した非機能要求項目については、考慮不要の理由をお客さまと合意しておく。</p> <p>※非機能要求グレードの「システム基盤の非機能要求に関する項目一覧」では、非機能要求を検討する上で品質やコストに大きな影響を与える項目を“重要項目”として定義している。非機能要求項目を選定する際には、RFPや提案書で明らかになっている非機能要求項目に加え、この重要項目を中心に検討すると良い。</p>	
上手に進めるためのポイント、注意事項	
モデルシステムの選定	
<p>➤ プロジェクトに完全一致するモデルシステムを選ぶ必要はない。プロジェクトにより近いモデルシステムを選び、そのマトリクスベース値がプロジェクト要求に合わない箇所注に注力検討することで、非機能要求全体の妥当性向上、非機能要求検討の効率向上をねらうことが、趣旨であることに留意する。</p>	
選択したモデルシステムと実現したいシステムの非機能要求の概要比較	
<p>➤ 「S3-02」以降のアクティビティを精緻に実施する前に、以後の非機能要求詳細検討に直接関わらない層(経営層、プロジェクトオーナー、業務部門や情報システム部門の組織長、等)を対象に、非機能要求グレードの大項目～中項目レベルで重視している項目の有無やその理由についてアンケートをとり、非機能要求の概要を体系的に把握すると良い。</p>	
「S3-02」以降の非機能要件定義アクティビティに関する共通ポイント	
<p>➤ 「S3-02」以降のアクティビティでは、お客さま側の機能要求検討担当者から非機能要求を詳細に聞き取り、経営層からの非機能要求の概要と矛盾がないかを確認する。</p> <p>➤ 非機能要件を定義する際に、想定していた実装方式の概要や方針を記録に残す。非機能要件の実現性を担保する実装方式や採用技術は、方式設計担当者と随時すり合わせながら、非機能要件定義を進める。</p>	

適用技法
<ul style="list-style-type: none">● 非機能要求グレード

S3-02 可用性要件の定義

サブプロセス目的
可用性は、システムを継続的に利用可能とするための要求である。いかにサービスを停止させないようにするか、影響範囲を極小化し、システムの稼働品質を保証するかを検討する。[2] なぜなら、システムは様々な要因により予期せぬサービス停止が発生するからである。
サブプロセス概要
システムの「継続性」、「耐障害性」、「災害対策」について検討し、要求されるレベルの指標を設定する。

S3-02-01 業務継続性

アクティビティ概要[2]	
<p>可用性を保証するにあたり、要求される業務の範囲とその条件を設定する。指標は、対象業務範囲、サービス切替時間、業務継続の要求度を使用する。</p> <ul style="list-style-type: none"> ✓ 対象業務範囲とは、稼働率を算出する際の対象範囲を指す。 ✓ サービス切替時間とは、想定できる障害（例えばハードウェアの故障等により業務が一時的に中断するケース）に対して、対策を施すこと（例えばクラスタ構成でのサーバの切替）により、業務再開までに要する時間を指す。 ✓ 業務継続の要求度とは、発生する障害に対して、どこまで業務を継続させる必要があるかを示す考え方の尺度を示している。システムを構成する機器や部位には、単一障害点(SPOF: Single Point Of Failure)が多数存在し、システム停止となるリスクを多く含んでいる。これらの SPOF を許容するか、冗長化などの対策で継続性をどこまで確保するかが要求の分かれ目となる。単一障害点とは、その単一箇所が働かないと、システム全体が障害となるような箇所を指す。 	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 組織一覧 ● 業務階層定義 ● 業務フロー ● システム要求一覧 ● システム機能一覧 	<ul style="list-style-type: none"> ● 可用性要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 可用性要求を明確にするには、「稼働している」と判断する状態の定義や、障害発生時の復旧目標を明らかにする必要がある。[2] ➤ 対象業務範囲(顧客向けフロント業務、バックエンド業務、等)毎に継続性レベルが異なる場合は、対象業務範囲毎にサービス切替時間、業務継続の要求度を定義する。 ➤ 業務停止による社会的影響、経済的損失の観点から、業務継続性要件を設定する。設定理由や根拠を明確にしておくことも重要。交通機関制御、金融業務、電子商取引、受注業務、出荷業務など、業務停止による多額の損失、顧客への影響、社会的信用失墜が想定される場合は高い業務継続性が求められる。 	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 高い業務継続性を求めず、長い業務中断を許容する場合、システムで自動化された復旧対策が採用されず、開発コストは下がるが、運用コストが増大するケースがあるので、注意する。 ➤ 連携先となる外部システムの可用性も考慮する。構築対象システムの可用性が、外部システムの可用性よりも劣る場合、外部システムに影響を与えないようにするために、可用性の対応レベルを引き上げる必要がないかを検討する。 	
適用技法	
<ul style="list-style-type: none"> ● リスク分析 ● ビジネスインパクト分析(BIA) 	

S3-02-02 目標復旧水準

アクティビティ概要[2]	
障害発生時の復旧目標を明らかにするため下表の項目を設定する。	
確認項目	説明
業務停止時	業務停止を伴う障害が発生した際、何をどこまで、どれ位で復旧させるかの目標を設定する。 指標は、RPO(目標復旧地点)、RTO(目標復旧時間)、RLO(目標復旧レベル)を使用する。
大規模災害時	大規模災害が発生した際、どれ位で復旧させるかの目標を設定する。指標はシステム再開目標を使用する。 大規模災害とは、火災や地震などの異常な自然現象、あるいは人為的な原因による大きな事故、破壊行為により生ずる被害のことを指し、システムに甚大な被害が発生するか、電力などのライフラインの停止により、システムをそのまま現状に修復するのが困難な状態となる災害をいう。
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務フロー ● システム要求一覧 	<ul style="list-style-type: none"> ● 可用性要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
業務停止時 <ul style="list-style-type: none"> ➤ RPO:いつのデータまで戻すか、RTO:どのくらいの時間で復旧させるか、RLO:何を復旧させるか、を具体的に定義する。 より新しいデータに、より早く、より広範囲に復旧させるにつれコストは上昇するので、業務視点から各メトリクス値の適切さを判断する。 ➤ RLO で業務復旧まで指定する場合、RTO が示す時間は、RLO が示す業務に必要なデータの復旧までを対象としたものになる。 実際の業務再開にかかる時間は、RLO 範囲外で復旧に必要な事項を加味して設定する必要がある。 ➤ 業務継続対策を前提とした「S3-02-01 業務継続性」のサービス切替時間と異なり、RTO は業務継続対策が無く業務停止した状態を基点として RPO,RLO を満たすまでの時間を指す。 	
大規模災害時 <ul style="list-style-type: none"> ➤ 大規模災害に対しては、システム再開目標として大まかな目標復旧地点と復旧時間と目標復旧レベルを設定する。 目標復旧レベルについては、業務停止時の目標復旧水準(RLO)を参考に検討する。 ➤ 大規模災害時について、お客さまが高度な RPO,RTO,RLO 要求を求める場合は、DR(ディザスタリカバリー)サイトの構築提案も検討する。 	
適用技法	
<ul style="list-style-type: none"> ● リスク分析 	

S3-02-03 稼働率

アクティビティ概要[2]	
<p>明示された利用条件の下で、システムが要求されたサービスを提供できる割合を稼働率として設定する。 利用条件は、「S3-04-01 保守運用」の運用時間や、「S3-02-02 目標復旧水準」の RLO、などで定義する。 その運用時間の中で、サービス中断が発生した時間により稼働率を求める。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● システム要求一覧 ● システム機能一覧 	<ul style="list-style-type: none"> ● 可用性要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ 一般的には、高い稼働率を目指すほどコストも上がる、「稼働率とコストはトレードオフの関係」にある。 むやみに高い稼働率を設定せず、対象業務で現実的に許容できる停止時間を検討し稼働率を設定する。 同じ業務ドメイン、業務範囲の他システムで設定した稼働率を参考にすることも有効。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-02-04 災害対策

アクティビティ概要	
<p>「S3-02-02 目標復旧水準」で定義した地震、水害、テロ、火災などの大規模災害時の目標復旧水準を満たし、業務継続性を実現するための要求を設定する。</p> <p>指標は、システムの復旧方針、データの外部保管方法、保管場所分散度、付帯設備の災害対策範囲を使用する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 組織一覧 ● 業務フロー ● 業務ルール定義 ● システム要求一覧 	<ul style="list-style-type: none"> ● 可用性要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 業務継続の重要度、想定する災害、災害発生時の業務影響度を分析し、システムの復旧方針を検討する。</p> <p>➤ DR サイトを構築する場合、RPO を実現するためのデータ反映手段、RTO を実現するシステム構成、運用体制、被災時に必要な業務とそのレベル、システムを利用できない業務の代替手段を検討する。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ システムの復旧方針では大規模災害への備えとして、代替機器の要否や、配置する機器、配置場所を決める。</p> <p>配置する機器については、本番環境との構成内容の差異を明確にする。</p> <p>➤ 付帯設備については、「S3-10-04 耐震/免震レベル」で定義する耐震震度が、災害対策の一部となるので、合わせて確認する。</p>	
適用技法	
<ul style="list-style-type: none"> ● リスク分析 ● ビジネスインパクト分析 	

S3-02-05 耐障害性

アクティビティ概要

耐障害性では、障害発生時においても要求されたサービスを維持するために、下表の項目を設定する。
耐障害性は冗長化により高まる。冗長化とは機器の一部に何らかの障害が発生した場合でも、システムの機能を維持し続けられるように、平常時から予備機器をバックアップとして配置し運用しておくことである。

冗長化の設定は、機器レベル、機器内のコンポーネントレベル、に分けて実施する。機器レベルは筐体を複数用意することによる冗長化、コンポーネントレベルは筐体を構成する部品（ディスク、電源、FAN、ネットワークカード等）を複数用意することによる冗長化を指す。

確認項目	説明
サーバ	サーバで発生する障害に対して、機器とコンポーネントの冗長化を設定する。
端末	端末で発生する障害に対して、機器とコンポーネントの冗長化を設定する。
ネットワーク機器	ルータやスイッチなどネットワークを構成する機器で発生する障害に対して、機器とコンポーネントの冗長化を設定する。
ネットワーク	ネットワークの信頼性を向上させるため、回線と経路の冗長化を設定する。同時に、セグメント分割の検討を行う。
ストレージ	ディスクアレイなどの外部記憶装置で発生する障害に対して、機器とコンポーネントの冗長化を設定する。

インプット

- 業務要求一覧
- システム要求一覧

アウトプット

- 可用性要件定義
- 非機能要求グレード表（確認項目反映済）

手順

—

【重要】上手く進めるためのポイント、注意事項

- 全ての機器、コンポーネントを冗長化すれば耐障害性は高まるがコストも増加する。
「S3-02-01 業務継続性」を満たすために必要なレベル設定を行う。

上手く進めるためのポイント、注意事項

- 仮想化技術の適用により、同一ハードウェア上にサーバ機能を集約させることで、冗長化に必要なハードウェア所要量を削減することも可能である。
- DR サイトを設置する際、許容する本番環境とのデータ同期デレイは重要な検討要素となる。
（日次同期で最悪 24 時間巻き戻っていいのか、3 分以上のデータロストを許さないのか、など）
DR サイトとの完全同期を保証すると 2 フェーズコミット等の対策が必要となり、システムの複雑化、オンラインレスポンス悪化などのリスクが高まる。

適用技法

- （特になし）

S3-03 性能・拡張性要件の定義

サブプロセス目的[2]
<p>システムの性能と将来のシステム拡張に関する要求を明確にする。</p> <p>要求が不明確な場合、「レスポンスが遅く実用に耐えない」、「必要以上に高性能なシステム基盤が導入される」など、適切な投資対効果が得られないため重要である。また、システムは一度構築すると長い期間使われることが多く、この間に利用者数や格納するデータ量が増えることでシステムのリソースが不足し、システムが本来の役割を果たせなくなることがある。この対策として拡張性を考慮する必要がある。</p>
サブプロセス概要
<p>サービスを提供する際リソースが効率よく使えるかを示すものが性能であり、主にレスポンスやスループットで要求を表現する。</p> <p>レスポンスはサービスを受ける側が要求を出してからサービスを受けられるまでの時間、スループットは単位時間当たり受けられるサービス量、処理量を示している。</p> <p>拡張性の代表的な例には、個々のリソースをより大きなものに入れ替えること(スケールアップ)、サーバや機器を追加すること(スケールアウト)がある。</p> <p>要件定義時に見込んでおくことで、機器増設の計画が立てやすく、あらかじめ余裕のある機器構成とすることで、増設時のコストを低くすることが可能である。</p>

S3-03-01 業務処理量

アクティビティ概要	
<p>性能目標値やリソース拡張性を決める上で、事前の必須確認項目である業務量と変動見通しを定義する。</p> <p>システム稼働開始時の通常の業務量を基準とし、ライフサイクル終了までの間で最大になる時点の増大率を想定し、合意する。</p> <p>システム稼働開始時の通常の業務量を設定する際の指標は、ユーザー数、同時アクセス数、データ量、オンラインリクエスト件数、バッチ処理件数、業務機能数を使用する。</p> <p>業務量増大度を設定する際の指標は、ユーザー数、同時アクセス数、データ量、オンラインリクエスト件数、バッチ処理件数、業務機能数のそれぞれの増大率を使用する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● データフロー ● 業務フロー ● イベント一覧 ● 業務ルール定義 ● システム要求一覧 ● システム機能一覧 	<ul style="list-style-type: none"> ● 性能・拡張性要求定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>通常時の業務量の調査</p> <hr/> <p>➤ それぞれの指標において、単一の値だけでなく、前提となる単位時間、時間帯や季節の特性なども考慮する。</p> <p>業務増大度の調査</p> <hr/> <p>➤ 今後のビジネス拡大やシステム導入拠点拡大を踏まえた、将来の想定業務量をお客さまと合意することが重要である。必要に応じ、システム稼働開始日の平均値や、開始後の定常状態との比較を行う場合もある。</p>	
上手く進めるためのポイント、注意事項	
<p>業務増大度の調査</p> <hr/> <p>➤ 通常時の業務量による性能・拡張性の検討のみでは、データ量の増加によるディスク追加発生や、オンライントランザクション件数・バッチ処理件数の増加によるサーバ能力不足、等の発生リスクがあるため、業務増大度を考慮する必要がある。業務量増大度が分からない場合は、分からない理由や対応方針、決定時期等を明確にし、余裕を持った仮設定値で性能目標値やリソース拡張性等の他のリスクの検討を進める。</p>	
適用技法	
<ul style="list-style-type: none"> ● 回帰分析 	

S3-03-02 性能目標値

アクティビティ概要	
<p>システム化する対象業務の特性をふまえた性能値と遵守率を、通常時・ピーク時、サブシステム・機能、などで分割した設定対象毎に設定する。 性能値は、「ターンアラウンドタイム」「レスポンスタイム」「スループット」「帳票出力枚数」などのメトリクスを使用して設定する。 指標は通常時性能・遵守度合い、ピーク時性能・遵守度合い、縮退時性能・遵守度合いを使用する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務フロー ● イベント一覧 ● システム要求一覧 ● システム機能一覧 ● システムフロー ● バッチ機能一覧 	<ul style="list-style-type: none"> ● 性能・拡張性要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 機能に求められる性能を業務の観点から判断し、機能性や複雑度とのバランスを取った性能目標値を設定する。 それらをする事なく、全機能に対して一律の性能目標を設定するのは非現実的である。 機能性重視の機能に対して、実現不可能な性能要求を設定することにつながり、開発終盤で、解決困難な性能問題に発展する可能性がある。 ➤ 性能目標値のシステムレイヤ上の対象範囲を明確にする。 Web システムオンライン機能のターンアラウンドタイムをメトリクスにする場合、リクエストを出しレスポンスが返るまでの間に含まれる、ブラウザ内処理やネットワーク伝送にかかる時間を測定対象に含めるのか、あるいはアプリケーションサーバ内の処理時間のみを対象にするのか、を決める。 原則、前者のような実行環境への依存が大きい要素は、測定対象から外すとよい。 ➤ 性能目標値を設定する前に「S3-03-01 業務処理量」を決め、ユーザー数、同時アクセス数、データ量、平常時/増大時、それらの想定時期を、性能目標値の前提として明示する。 	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-03-03 リソース拡張性

アクティビティ概要	
<p>ハードウェア、ネットワークリソースの拡張性は下表[2]の項目を設定する。</p> <p>拡張性の高い機器やサービスはコスト増となるため、業務増大度の調査結果に鑑み適切な拡張性を持たせる必要がある。</p>	
確認項目	説明
CPU 拡張性	<p>システム運用開始時の CPU 利用率と CPU スロットの空き具合から確認する。</p> <p>システム運用開始時の CPU 利用率が少ないほど拡張性はあるが、CPU のコストは高く、無駄が生じていることになる。</p> <p>CPU 搭載余裕有無は空きスロットの有無と空きスロット数を確認することで、拡張余力があるかどうかを示す。</p> <p>指標は CPU 利用率、CPU 搭載余裕有無を使用する。</p>
メモリ拡張性	<p>システム運用開始時のメモリ利用率とメモリスロットの空き具合から拡張性を確認する。</p> <p>メモリ利用率が少ないほど拡張性はあるが、メモリのコストは高く、無駄が生じていることになる。</p> <p>メモリ搭載余裕有無は空きスロットの有無と空きスロット数を確認することで、拡張余力があるかどうかを示す。</p> <p>指標はメモリ利用率、メモリ搭載余裕有無を使用する。</p>
ディスク拡張性	<p>システム運用開始時のディスク利用率とディスク増設スロットの空き具合から確認する。</p> <p>ディスク利用率が少ないほど拡張性はあるが、ディスクのコストは高く、無駄が生じていることになる。</p> <p>ディスク搭載余裕有無は空きスロットの有無と空きスロット数を確認することで、拡張余力があるかどうかを示す。ディスクは内蔵ディスクが不足しても外部増設が可能であり、CPU やメモリより拡張性は高い。</p> <p>指標はディスク利用率、ディスク増設利用有無を使用する。</p>
ネットワーク拡張性	<p>フロア内、拠点内と拠点間・社外接続にわけて、システム運用開始時の帯域状況および接続数から帯域と接続数の拡張性を確認する。指標はネットワーク機器設置範囲を使用する。</p>
サーバ処理能力増強	<p>将来の業務量増大に備える方法(スケールアップ/スケールアウト)をあらかじめ考慮し、最適な方法を確認しておく。どちらの方法を選択するかはシステムの特徴によって使い分けることが必要となる。</p> <p>指標はスケールアップ、スケールアウトを使用する。</p>
インプット	
● システム要求一覧	
アウトプット	
● 性能・拡張性要件定義	
手順	
—	
【重要】上手に進めるためのポイント、注意事項	
<p>➤ リソース拡張性を設定する前に「S3-03-01 業務処理量」を決め、システム稼働時の各種リソースの余裕率を把握する。</p> <p>➤ いずれの確認項目も、運用開始時の余裕率が低く、かつ増強の備えが無い場合、少しの業務量増大で機器やネットワーク等の増設が必要になる可能性がある。「S3-03-01 業務処理量」で決める業務処理量の増大率を前提としつつ、更にリソースに余裕を持たせる。</p>	
上手に進めるためのポイント、注意事項	
サーバ処理能力増強	
<p>➤ サーバはシステムの成長に合わせて計画的にスケールアップやスケールアウトができるような構成にしておくことが肝要である。</p> <p>➤ スケールアップは、より処理能力の大きなサーバとの入れ替えを行うことで処理能力の増強を行う。 オンライントランザクション処理のような更新系の割合が多いシステムでアプリケーションサーバをスケールアップする場合を想定している。</p> <p>➤ スケールアウトは同等のサーバを複数台用意し、サーバ台数を増やすことで処理能力の増強を行う。 Web サーバと負荷分散装置などフロント部分を複数台用意する場合を想定している。</p>	
適用技法	
● (特になし)	

S3-03-04 性能品質保証

アクティビティ概要	
<p>構築したシステムが想定した性能を発揮できるか測定する試験をどのように行うかを検討し、下表の項目[2]を設定する。</p> <p>性能テストでは、設定した性能目標値の実現度合いを評価するためのテストをどの程度行うかを定める。</p> <p>このテストが不十分であるとシステム稼働後処理能力に不足が生じ、システムが本来の役割を果たせなくなることがある。</p>	
確認項目	説明
性能テスト	<p>構築したシステムが当初/ライフサイクルに渡っての性能を発揮できるかのテストの測定頻度と範囲。</p> <p>測定頻度は、構築当初に測定するのか、運用中必要時に測定するのか、運用中定常的に測定するのかを設定する。</p> <p>指標は測定頻度、測定範囲を使用する。</p>
スパイク負荷対応	<p>通常時の負荷と比較して、非常に大きな負荷が短時間に現れることを指す。業務量の想定されたピークを超えた状態。</p> <p>特に B2C システムなどクライアント数を制限できないシステムで発生する。</p> <p>システムの処理上限を超えることが多いため、Sorry 動作を実装し対策する場合が多い。</p> <p>指標はランザクション保護を使用する。</p>
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● システム要求一覧 ● システム機能一覧 ● システムフロー 	<ul style="list-style-type: none"> ● 性能・拡張性要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>性能テスト</p> <hr/> <p>➤ 設計工程からシステムテスト工程に渡って各工程の検証ポイントを決め、性能目標達成へ向けた継続的な検証を行う必要がある。検証の積み上げなく、運用開始前のわずかな性能測定やチューニング等で性能目標を確保できると考えてはならない。テスト内容や実施時期は「S3-11-01 テスト工程定義と各テスト工程の役割分担」で主要要件を固め、全体テスト計画や工程別テスト計画で具体化する。</p> <p>➤ 高い性能を求められるサブシステム・機能は、構築時に加え、運用開始後も定常的に測定する事が望ましい。</p>	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-04 運用・保守要件の定義

サブプロセス目的
運用・保守要件の定義は、システムの運用と保守のサービスに関する要件、システムの運用方法や管理者の作業手順を決定する。
サブプロセス概要
<p>正常時、メンテナンス作業時、障害発生時のそれぞれの運用パターンに対する要求を検討する「通常運用」(※)、「保守運用」、「障害時運用」と、それらシステム運用を実現する環境や体制を検討する「運用環境」、「サポート体制」、「その他の運用管理方針」に関わる項目を検討、確認し定義する。[2]</p> <p>※:本書では、「通常運用」を下記3カテゴリに分割し、記載している</p> <ul style="list-style-type: none">①運用時間:「S3-04-01 保守運用」の「運用時間」に記載②バックアップ:「S3-06 バックアップ要件の定義」に記載③運用監視:「S3-05 運用監視要件の定義」に記載

S3-04-01 保守運用

アクティビティ概要	
<p>通常運用、保守運用の要件定義にあたり下表[2]の項目を確認、および設定する。</p> <p>システムを安定稼働させるために必要となる、メンテナンス作業の方針や内容に関する項目を設定する。</p> <p>これらは導入する機器の選定に大きく影響する。</p>	
確認項目	説明
運用時間	システム運用を行う時間。利用者やシステム管理者に対してサービスを提供するために、システムを稼働させ、オンライン処理やバッチ処理を実行している時間帯を確認する。
計画停止	点検作業や領域拡張、デフラグ、マスターデータのメンテナンス等、システムの保守作業の実施を目的とした、事前計画済みのサービス停止に関する項目を確認する。
運用負荷削減の検討	保守運用に関する作業負荷を削減するための設計に関する項目を確認する。 保守作業とは、保守運用に伴うシステム基盤を維持管理するための作業を指し、点検作業やパッチ適用等のアップデート作業、領域拡張、デフラグ、ログローテーション等を想定している。障害対応や復旧作業などは含まない。
パッチ適用ポリシー	パッチ情報の展開とパッチ適用のポリシーに関する項目を確認する。 パッチリリース情報の提供元、パッチの適用方針、パッチの適用タイミング、パッチ検証の実施有無について、お客さまとの役割分担を含め決定し、ポリシーを作成する。
活性保守	サービス停止の必要がない活性保守が可能なコンポーネントの範囲を確認する。 ハードウェア活性保守とは、システムを停止せずにハードウェア交換やファームウェア更新といった保守作業を実施することである。 ソフトウェア活性保守とは、システムを停止せずに OS やミドルウェア、アプリケーションのパッチ適用を実施することである（例：マルチサーバ環境におけるローリングアップグレードなど）。
定期保守頻度	システムの保全のために必要なハードウェアまたはソフトウェアの定期保守作業の頻度について確認する。
予防保守レベル	システム構成部品が故障に至る前に予兆を検出し、事前交換などの対応をとる保守の要否について確認する。 機器の場合、定期的に点検を行い、消耗が速い機器や部品は規定の利用期間や利用回数が経過した時点で正常に稼働していても交換するなどの対応を行う。 ソフトウェアの場合、潜在的な不具合を見つけ、障害として顕在化する前に修正することなどを意味する。
インプット	
<ul style="list-style-type: none"> ● 現行システム運用スケジュール ● 業務要求一覧 ● 業務フロー ● 業務ルール定義 ● システム要求一覧 ● バッチ機能一覧 	アウトプット
	<ul style="list-style-type: none"> ● 運用・保守要件定義 ● 非機能要求グレード表（確認項目反映済）
手順	
—	
【重要】上手に進めるためのポイント、注意事項	
<p>運用時間</p> <p>➤ 運用時間は、システムの可用性の実現レベルを表す項目でもあり、「S3-02-03 稼働率」と整合しているか確認する。</p> <p>計画停止</p> <p>➤ 計画停止の有無は、システムの可用性の実現レベルを表す項目でもあり、「S3-02-03 稼働率」と整合しているか確認する。</p> <p>パッチ適用ポリシー</p> <p>➤ セキュリティパッチについては「S3-09-02 セキュリティリスク分析」に基づき、別途適用範囲、適用方針、適用タイミングを合意する。</p>	
上手に進めるためのポイント、注意事項	

運用時間

- 運用時間は、業務要件を考慮し、オンライン/バッチ別にシステム稼働時間帯を決める。
- リポートや大量バックアップを行う休日/祝祭日や月初/月末など、通常とは異なる運用スケジュールが必要な場合は、「運用時間(特定日)」に別途定義する。またお客さま特有、ベンダー特有の休日も特定日として認識し、運用保守体制等を整合させる。

計画停止

- 計画停止有りの場合、事前のバックアップや、システム構成に応じた手順準備など、運用準備事項、成果物をお客さまと合意しておく。
- 計画停止無しの場合、無停止で保守作業が可能なシステム構成とするための対応およびコスト増加があることをお客さまに伝える。

運用負荷削減の検討

- JOB 管理システムは運用負荷削減に有効である。
既存の JOB 管理システムの有無を確認し存在する場合は、設計の前提としてよいか確認する。
- システム基盤の保守運用作業を自動化するためには、特別な運用管理ツールを導入や、さまざまな作り込みを実施する必要がある。
そのため導入コストは増大するが、お客さまが実施すべき保守運用作業が簡略化あるいはなくなると考えられるので、運用コストは減少する。

パッチ適用ポリシー

- リリースされるパッチが個別パッチであるか、集合パッチであるかによって、パッチの適用方針や適用タイミングが変わる場合は、個別に合意する必要がある。[2]
- セキュリティパッチの適用範囲は、OS、ミドルウェア等毎に確認する。
セキュリティパッチの適用を検討する際には、システム全体への影響を確認し、パッチ適用の可否を判断する。
影響の確認等については保守契約の内容として明記されることが望ましい。
- セキュリティパッチには、ウィルス定義ファイル等を含む。

活性保守

- 24 時間 365 日無停止の業務に関わるコンポーネント等は、活性保守が必要となる。
活性保守には、ローリングアップグレードなどの仕組み・手順を整備するコストがかかる。
お客さまが活性保守を行わないと判断した場合は、お客さまに 24 時間 365 日無停止の要求を取上げて頂き、停止条件や時間を合意しておく。

定期保守頻度

- ハードウェアの定期保守点検は必要である前提で考える。
- 業務的に保守点検が可能な時期・時間について調査の上で合意に導く。

予防保守レベル

- 部材の平均耐用年数および故障率と、業務の重要性や故障発生時の想定業務影響から、予防保守要否やレベルを検討し、合意する。

適用技法

- (特になし)

S3-04-02 障害時運用

アクティビティ概要	
<p>障害時運用の要件定義にあたり下表[2]の項目を設定する。</p> <p>システムに障害が発生した場合、速やかに復旧させるための対応方法、内容を定義する。</p> <p>導入機器や要員・部材の確保等、コストに大きく影響する項目が多いため、可用性の観点とあわせて検討する。</p>	
確認項目	説明
復旧作業	業務停止を伴う障害が発生した際の復旧作業に必要な労力。
障害復旧自動化の範囲	障害復旧に関するオペレーションを自動化する範囲。
システム異常検知時の対応	システムの異常を検知した際のベンダー側対応。 ベンダー側対応の要否、ベンダー側 SE 対応可能時間帯、SE 駆けつけ到着時間、SE 到着平均時間を検討する。
交換用部材の確保	障害の発生したコンポーネントに対する交換部材の確保方法。 保守部品確保レベル、予備機の有無等、当該システムに関する保守部品の確保レベルについて検討する。
インプット	アウトプット
<ul style="list-style-type: none"> ● 現行システム運用スケジュール ● 業務要求一覧 ● 業務フロー ● システム要求一覧 ● システム機能一覧 ● システム機能俯瞰図 	<ul style="list-style-type: none"> ● 運用・保守要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>復旧作業</p> <hr/> <p>➤ 選定したレベルに応じて、お客さま側・ベンダー側それぞれの体制や権限の整理を実施し、その役割を明確にする。</p>	
上手く進めるためのポイント、注意事項	
<p>復旧作業</p> <hr/> <p>➤ 復旧用製品(バックアップ・リカバリを行う製品)による復旧を行う場合、自動化範囲(自動リカバリ機能充足率など)を定義するが、その前に、コスト・可用性に大きく影響する、復旧用製品の要否の観点で要件を整理する。</p> <p>➤ 障害によりシステムが復旧不可能となった場合に、代替業務でカバーすることが可能な運用手段(代替機あるいは人手による運用)の有無および要否について検討する。</p>	
<p>障害復旧自動化の範囲</p> <hr/> <p>➤ 障害復旧作業を自動化するためには、障害のパターン毎に複雑な判断を行うスクリプトを作成する必要があり開発コストが増大する。一方、障害発生時の復旧作業が迅速化され、ミスも少なくなるため運用コストは減少する。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-04-03 運用環境

アクティビティ概要	
<p>運用環境の要件定義にあたり下表[2]の項目を設定する。</p> <p>システム開発からテスト、運用に至る過程で必要な環境の準備、また運用に必要なマニュアルの準備のレベルなどを検討する。</p> <p>これらはシステムの運用に大きく影響するものであり、漏れが無いように注意する必要がある。</p>	
確認項目	説明
開発用環境の設置	お客さままたはベンダーがシステムに対する開発作業を実施する目的で導入する環境について定義する。 開発用環境とは、本番環境とは別に開発専用を使用することのできる機材一式のことを指す。
試験用環境の設置	お客さまがシステムの動作を試験する目的で導入する環境について定義する。 試験用環境とは、本番環境とは別に試験専用を使用することのできる機材一式のことを指す。
マニュアル準備レベル	運用のためのマニュアルの準備のレベルを確認する。 通常運用のマニュアルには、システム基盤に対する通常時の運用(起動・停止等)にかかわる操作や機能についての説明が記載される。 保守運用のマニュアルには、システム基盤に対する保守作業(部品交換やデータ復旧手順等)にかかわる操作や機能についての説明が記載される。 障害発生時の一次対応に関する記述(系切り替え作業やログ収集作業等)は通常運用のマニュアルに含まれる。 バックアップからの復旧作業については保守運用のマニュアルに含まれるものとする。
リモートオペレーション	システムの設置環境とは離れた環境からのネットワークを介した監視や操作の可否を確認する。 監視の内容については、「S3-05 運用監視要件の定義」で確認する必要がある。
インプット	アウトプット
<ul style="list-style-type: none"> ● 現行開発環境マニュアル ● 現行試験環境マニュアル ● 現行運用マニュアル ● 現行保守マニュアル ● 現行ネットワーク図 ● 現行リモート接続手順書 ● 現行システム機能俯瞰図 ● 業務要求一覧 ● 業務フロー ● システム要求一覧 ● システム機能俯瞰図 ● 外部IF 機能要件定義 	<ul style="list-style-type: none"> ● 運用・保守要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手に進めるためのポイント、注意事項	
—	
上手に進めるためのポイント、注意事項	
試験用環境の設置	
<p>➤ 本番環境と同等の試験用環境を設置できる事が理想だが、コスト制約や外部システム接続での物理制約と調整して、決定する。 ただし、保守や保守開発で行う試験工程の前提環境を定義し、試験用環境でそれらをカバーすることは必須。</p> <p>➤ 外部システムと連携する必要がある場合は、試験環境に外部接続環境を準備するかどうか、その要否を検討する。 検討にあたっては以下の点を考慮する。</p> <ul style="list-style-type: none"> ✓ 外部システム利用の責任の所在を明確にする。(お客さま/ベンダー) ✓ 外部システムの利用規約、仕様を確認する。 ✓ セキュリティの確保(利用者アクセス権限など) 	

マニュアル準備レベル

- ユーザーの運用に合わせカスタマイズしたマニュアルは、作成するためにコストがかかるため導入コストが増大するが、ユーザーが運用時に手順を調査する負担が減少するため運用コストは減少する。[2]
導入コストと運用コストのバランスを取って、マニュアルのお客さま個別対応の度合いを決定する。

リモートオペレーション

- リモート監視やリモート操作を実装するためには、特別なハードウェア・ソフトウェアを導入する必要があり導入コストが増大する。しかし、運用状況の確認のために管理者がわざわざサーバの設置場所まで移動する必要がなくなるため、運用コストは減少する。[2]
導入コストと運用コストのバランスを取って、リモート監視やリモート操作の要否、対象範囲を決定する。

適用技法

- （特になし）

S3-04-04 サポート体制

アクティビティ概要	
<p>サポート体制の要件定義にあたり下表[2]の項目を設定する。</p> <p>システムの運用・保守にあたっては、その契約に関する事項、システム運用に関するお客さま・ベンダー間の役割分担に関する事項、ベンダーのサポート体制に関する事項を定義する。これらはシステムの運用品質を確保するために事前に検討しておくべき事項である。</p>	
確認項目	説明
保守契約(ハードウェア・ソフトウェア)	保守が必要な対象ハードウェア・ソフトウェアの範囲を確認する。 複数のベンダーの製品から構成されるシステムに対してワンストップのサポート窓口となる場合は、システム全体に対するサポートサービスを提供するベンダーと契約を行う事となる。
ライフサイクル期間	運用保守の対応期間および、実際にシステムが稼働するライフサイクルの期間を確認する。 ここでのライフサイクルとは、次回システム更改までの期間を指す。
メンテナンス作業役割分担	メンテナンス作業に対するお客さま/ベンダーの役割分担、配置人数の確認を行う。
一次対応役割分担	障害発生時に業務再開を最優先とした暫定対応等を行う、一次対応のお客さま/ベンダーの役割分担、一次対応の対応時間、配備人数の確認を行う。
サポート要員	サポート体制に組み入れるベンダー要員の人数や対応時間、スキルレベルの確認を行う。
導入サポート	システム導入時の特別対応の有無および期間の確認を行う。 ユーザー受入テスト時のサポート体制・期間と、本稼働時のサポート体制・期間に分けて確認する。
オペレーション訓練	お客さま向けのオペレーション訓練実施の要否、お客さま/ベンダーの役割分担、実施範囲、実施頻度について確認する。
定期報告会	保守に関する定期報告会の開催の要否、開催頻度と報告内容のレベルを確認する。
インプット	
<ul style="list-style-type: none"> ● ソフトウェア構成図 ● ミドルウェア構成図 ● 現行運用保守契約書 ● 現行オペレーションマニュアル ● 現行定期報告会資料 ● 業務要求一覧 ● 業務フロー ● システム要求一覧 	アウトプット
	<ul style="list-style-type: none"> ● 運用・保守要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>保守契約(ハードウェア・ソフトウェア)</p> <hr/> <p>➤ 必ずしもシステム開発を行ったベンダーがワンストップのサポート窓口となる必要はなく、外部のベンダーやお客さまの契約にした方がよい場合も有り得る。 その際はシステム開発を行ったベンダーが保守担当となるアプリケーションに関わるソフトウェア類の希望サポート条件を明らかにし提示する。</p> <p>➤ ハードウェア・ソフトウェアの保守可能期間が後述するライフサイクル期間よりも短い場合は、保守契約延長や保守可能バージョンへのバージョンアップ等の対応が必要となる。[2]</p>	
<p>ライフサイクル期間</p> <hr/> <p>➤ 自社、他社問わず、システムを構成する、製品・サービスの保守可能期間を確認しておく。</p>	

メンテナンス作業役割分担

- お客さまがメンテナンス作業の一部あるいは全てを行う場合、お客さまに必要スキル・知識を持って頂く必要があるため、現状の保有状況を確認しておくとい。当該スキル・知識が不足している場合、開発作業期間内でOJTを行う等の開発計画上の考慮が必要になる可能性がある。

一次対応役割分担

- 障害発生の一対応を行う要員は業務やシステムに関する知識、情報が必要となる。
しかし、それらを持った開発担当者が引き続き運用・保守を担当するとは限らないため、運用・保守に引き継ぐ開発時文書や保守運用のマニュアルの整備計画を、プロジェクト計画に含めることを検討する。

サポート要員

- エスカレーション対応が必要となる製品がある場合は、エスカレーション先、待機方法(オンサイト、自社待機など)や連絡方法を決定する。[2]

適用技法

- (特になし)

S3-04-05 その他の運用管理方針

アクティビティ概要	
<p>その他の運用管理方針の要件定義にあたり下表[2]の項目を設定する。 内部統制対応、サービスデスク、インシデント管理などのITIL(*1)に関する事項を定義する。 これらについて、管理することをお客さまとベンダーが合意した際には、合わせて実現方針について確認する必要がある。 (*1) ITIL(Information Technology Infrastructure Library)とはITサービスマネジメントにおけるベストプラクティスをまとめた体系的なガイドラインである。</p>	
確認項目	説明
内部統制対応	運用プロセスの内部統制(システム運用の妥当性)対応の実施要否について確認を行う。 ここでの内部統制とは、運用管理において統一したポリシーを作成・適用し、それに基づいた正しい作業内容や手順を定めて、規定通りの作業を行うような体制作りを行うことを指す。
サービスデスク	ユーザーからの問合せに対する、単一の窓口機能の提供要否を確認する。 窓口機能を提供する場合、サービス時間、サービスメニュー等について確認する。
インシデント管理	通常業務を遂行できない何らかの状態(インシデント)を迅速に回復させるプロセスを実施するかどうかを確認する。
問題管理	インシデントの根本原因を追究し、取り除くための処置を講じるプロセスを実施するかどうかを確認する。
構成管理	ハードウェアやソフトウェアなどのIT環境の構成を適切に管理するためのプロセスを実施するかどうかを確認する。
変更管理	IT環境に対する変更を効率的に管理するためのプロセスを実施するかどうかを確認する。
リリース管理	ソフトウェア、ハードウェア、ITサービスに対する実装を管理するためのプロセスを実施するかどうかを確認する。
インプット	
<ul style="list-style-type: none"> 社内規定 現行運用マニュアル 現行オペレーションマニュアル 現行運用保守契約書 業務要求一覧 業務フロー 	<ul style="list-style-type: none"> 運用・保守要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 各確認項目の領域は、運用プロセスとして連続性がある。 運用プロセス俯瞰図で、各領域の範囲と領域間の関係を示すことで、お客さまと運用プロセスの全体像を共有しやすくなる。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ 確認項目ごとに、既存プロセスに従うのか、新規プロセスを規定するのかが確認する。 新規プロセスを規定する場合、主要要件の抽出と実施概要を定める程度に留め、具体的な内容は後続の運用方式設計(要件定義フレームワーク範囲外)で検討する。</p> <p>➤ 構成管理、インシデント管理、問題管理等のツール適用を検討する。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-05 運用監視要件の定義

サブプロセス目的
システム全体、あるいはそれを構成するハードウェア・ソフトウェア(業務アプリケーションを含む)に対する監視項目、監視レベルの確認と設定を行う。
サブプロセス概要
<p>監視要件のグレードは以下の3層に大別される。</p> <ul style="list-style-type: none">① ハードウェアやソフトウェアの各種ログを用いて死活監視を行う② アプリケーションの各業務機能が正常に稼働しているかどうか監視を行う。③ 性能やリソース使用状況まで監視し、障害の予兆検出を行う。 <p>システム運用上の重要度により監視対象を定め監視対象毎に監視情報、監視レベルを設定する。</p>

S3-05-01 監視情報

アクティビティ概要	
<p>監視とは、情報収集を行った結果に応じて適切な宛先に発報することを意味する。 ここでは、運用に関する監視内容、監視間隔、監視を行う監視システムの有無、の確認も行う。(不正監視は「S3-09-07 不正監視」で検討)</p> <p>① 監視情報のレベル[2] 運用に関して、何を対象に、何を監視し、発報するのか、監視情報のレベルを設定する。監視情報のレベルは以下の4層に大別される。</p> <ul style="list-style-type: none"> ✓ 死活監視 対象のステータスがオンラインの状態にあるかオフラインの状態にあるかを判断する監視である。 ✓ エラー監視 対象が出力するログ等にエラー出力が含まれているかどうかを判断する監視のこと。トレース情報を含む場合は、どのモジュールでエラーが発生しているのか詳細についても判断することができる。 ✓ リソース監視 対象が出力するログや別途収集するパフォーマンス情報に基づいてCPUやメモリ、ディスク、ネットワーク帯域といったリソースの使用状況を判断する監視である。 ✓ パフォーマンス監視 対象が出力するログや別途収集するパフォーマンス情報に基づいて、業務アプリケーションやディスクI/O、ネットワーク転送等の応答時間やスループットについて判断する監視である。 <p>② 監視間隔[2] 「監視情報のレベル」で決めた監視内容を、実行する頻度を監視間隔として設定する。監視間隔のレベルは以下の6層に大別される。</p> <ul style="list-style-type: none"> ✓ 監視を行わない ✓ 不定期監視(手動監視) ✓ 定期監視(1日間隔) ✓ 定期監視(数時間間隔) ✓ リアルタイム監視(分間隔) ✓ リアルタイム監視(秒間隔) <p>③ 監視システムの有無 監視を実行する監視システムについて、既存利用するか、新規導入するかを確認する。 監視システムによる監視情報の検知可否と監視間隔は、「S3-05-02 ソフトウェアに対する監視」項以降で確認する。 また監視システムを利用しない場合は、代替手段(手動監視または監視システムの設計・実装)を検討する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 現行監視設定書 ● 現行監視手順書 ● システム要求一覧 ● システム機能一覧 ● システムフロー 	<ul style="list-style-type: none"> ● 監視要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 監視情報のレベル、監視間隔は、ソフトウェア/ハードウェア/ネットワークの監視対象ごとに設定する。 システム全体で一律に設定するものではない。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ 監視システムの構築・導入にはコストが発生するがエラー監視やリソース監視、パフォーマンス監視を行うことによって、 障害原因の追究が容易となる、障害を未然に防止できる、などシステムの品質を維持するための運用コストが下がる。[2]</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-05-02 ソフトウェアに対する監視

アクティビティ概要	
<p>ソフトウェア(*1)に対する監視の要件定義にあたり、下表[2]の項目を設定する。</p> <p>システム全体、プロセス、データベースの各ソフトウェアが正しく機能しているかの監視レベルを設定する。</p> <p>(*1)オンプレミスで仮想マシンがある場合は、これをソフトウェアとして監視対象とする。</p>	
確認項目	説明
システムレベルの監視設定	業務アプリケーションも含め、そのシステムを構成する複数のサーバ等の状態確認結果から、システムとして機能する状態にあるかどうかを判断する。 バックアップの監視やバッチジョブのステータスの監視などが該当する。
プロセスレベルの監視設定	アプリケーションやミドルウェア等のプロセスが正しく機能しているかどうかを判断する。 主に OS コマンドによるプロセスの情報(死活、CPU 使用率、メモリ使用率など)を監視する。
データベースレベルの監視設定	DBMS の機能として提供される情報を確認し、正しく機能しているかを判断する。 ログ出力内容やパラメータ値、ステータス情報、領域使用率等の監視を行う。
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 システム機能一覧 システムフロー 	<ul style="list-style-type: none"> 監視要件定義非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>システムレベルの監視設定</p> <hr/> <p>➤ 監視を行う場合は、システムレベルについての監視情報と監視間隔を個別に確認する。[2] 例えばシステムが提供するいくつかの機能のうち、重要度の高い一部の機能のみを対象に監視を行う事も想定した上で合意に導く。</p> <p>プロセスレベルの監視設定</p> <hr/> <p>➤ 監視を行う場合は、プロセスレベルについての監視情報と監視間隔を個別に確認する必要がある。[2] 例えばシステム上で稼働する複数のプロセス(アプリケーションおよびミドルウェア)のうち、重要度の高い一部のプロセスのみを対象に監視を行う事も想定した上で合意に導く。</p> <p>データベースレベルの監視設定</p> <hr/> <p>➤ 監視を行う場合は、データベースレベルについての監視情報と監視間隔を個別に確認する必要がある。[2] 例えばシステム上で稼働する複数のデータベースのうち、重要度の高い一部のデータベースのみを対象に監視を行う事も想定した上で合意に導く。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-05-03 ハードウェアに対する監視

アクティビティ概要	
<p>ハードウェアに対する監視の要件定義にあたり、下表[2]の項目を設定する。 ストレージ、サーバ(ノード)、端末/ネットワーク機器の各ハードウェアが正しく機能しているかの監視レベルを設定する。</p>	
確認項目	説明
ストレージレベルの監視設定	ディスクアレイ等の外部記憶装置に関して、状態を確認し、正しく機能しているかを判断する。 OS コマンドによって確認できるディスク使用率等の他、ファームウェアが出力するログ情報などの監視を行う。
サーバ(ノード)レベルの監視設定	対象のサーバが OS レベルで正しく機能しているかを判断する。 ハートビート監視などが該当する。
端末/ネットワーク機器レベルの監視設定	クライアント端末やルータ等のネットワーク機器に関して、状態を確認し、正しく機能しているかを判断する。 ハートビート監視の他、個別のファームウェア等が出力する情報に基づく監視が該当する。
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> 監視要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>ストレージレベルの監視設定</p> <hr/> <p>➤ 監視を行う場合は、ストレージレベルについての監視情報と監視間隔を個別に確認する必要がある。[2] 例えば、システムに接続される複数のストレージのうち、重要度の高い一部のストレージのみを対象に監視を行う事も想定した上で合意に導く。</p>	
<p>サーバ(ノード)レベルの監視設定</p> <hr/> <p>➤ 監視を行う場合は、サーバ(ノード)レベルについての監視情報と監視間隔を個別に確認する必要がある。[2] 例えばシステム上に存在する複数のサーバ(ノード)のうち、重要度の高い一部のサーバ(ノード)のみを対象に監視を行う事も想定した上で合意に導く。</p>	
<p>端末/ネットワーク機器レベルの監視設定</p> <hr/> <p>➤ 監視を行う場合は、端末/ネットワーク機器レベルについての監視情報と監視間隔を個別に確認する必要がある。[2] 例えば、システム上に存在する複数の端末/ネットワーク機器のうち、重要度の高い一部の端末/ネットワーク機器のみを対象に監視を行う事も想定の上で合意に導く。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-05-04 ネットワークの監視

アクティビティ概要	
<p>ネットワーク・パケットレベルの監視は、ネットワーク上を流れるパケットの情報を基に、正しく機能しているのかを確認する事である。[2] ここでは監視範囲・監視対象情報を確認する。パケットロスやネットワーク帯域の使用率などの監視などが該当する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> 監視要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ 監視を行う場合は、ネットワーク・パケットレベルについての監視情報と監視間隔を個別に確認する必要がある。[2] 例えばシステム上の複数のネットワーク経路のうち、重要度の高い一部のネットワーク経路のみを対象に監視を行うことも想定の上で合意に導く。</p> <p>➤ WAN、LAN 双方で例えば以下のようなものが対象となる。</p> <ul style="list-style-type: none"> ✓ セキュリティ的不正パケット ✓ サーバ等の死活確認 ✓ 冗長化モニタリング ✓ 業務流量管理 ✓ キャパシティ管理 等 	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-06 バックアップ要件の定義

サブプロセス目的
データにアクセス出来ないことによる、機会損失と復旧コストを最小限に抑えるために実施すべき対策を定義する。 データ紛失・破損のリスクは常に存在し、データ保全のためにバックアップ処理は必要不可欠な行為である。
サブプロセス概要
システムの規模や用途により、適切な範囲と頻度でバックアップ運用を行う必要がある。バックアップ対象のデータ、バックアップ頻度、保存期間、等を、システムやデータの重要度、運用や維持のコスト、その他の要因から総合的に判断して、バックアップの計画を立て運用する。

S3-06-01 データ復旧範囲

アクティビティ概要	
データ復旧範囲は、一部のデータのみとするのか、システムの全データとするのかを設定する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務フロー ● 業務ルール定義 ● システム要求一覧 ● システム機能俯瞰図 ● システムフロー ● 外部IF 機能要件定義 ● バッチ機能一覧 ● 論理データモデル定義 	<ul style="list-style-type: none"> ● バックアップ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ システムを障害から復旧するためには、データバックアップ以外に、OS やアプリケーションの設定ファイル等を保管するシステムバックアップも必要となることが考えられる。システムバックアップの取得方法や保管方法についても、同時に検討しなければならない。 ➤ 復旧範囲を「一部データ」とする場合は、「S3-02-01 業務継続性」に定める業務継続性要件を満たすよう、対象データを選定する。 ➤ 外部接続先システムからのデータ授受で復旧可能なデータがないか確認するとよい。バックアップ設計、運用を削減できる。 	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-06-02 バックアップ利用範囲

アクティビティ概要	
バックアップデータの利用目的を設定する。障害発生時のデータ損失防止のためなのか、作業ミスなどによって発生したデータ損失を回復したいのか、内部統制対応でデータの履歴を保存する必要があるのかを判断する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務フロー ● 業務ルール定義 ● システム要求一覧 ● システム機能俯瞰図 ● システムフロー ● 外部IF 機能要件定義 ● バッチ機能一覧 ● 論理データモデル定義 	<ul style="list-style-type: none"> ● バックアップ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
➤ ユーザーエラーからの回復の場合、システムとしては正常に完了してしまった処理を元に戻さなければならないため、複数世代のバックアップの管理や時間指定回復(Point in Time Recovery)等の機能が必要となる場合が考えられる。[2]	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-06-03 バックアップ自動化の範囲

アクティビティ概要	
バックアップ実行に必要な作業ステップ(※)のうち、自動化する対象を設定する。 ※例(バックアップジョブ起動、バックアップ対象選択、バックアップ先メディア選択、データ転送、etc)	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務フロー ● システム要求一覧 ● システム機能俯瞰図 ● システムフロー ● 外部IF 機能要件定義 ● バッチ機能一覧 ● 論理データモデル定義 	<ul style="list-style-type: none"> ● バックアップ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
➤ バックアップ運用の自動化を実現するためには、ハードウェア・ソフトウェアに対する投資が必要となり導入コストは増大する。 しかし、運用中におけるバックアップ作業をお客さまが実施する必要がなくなるため、その分運用コストは減少すると考えられる。[2] 全体的なコストバランスを考えて決定する。	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-06-04 バックアップ取得間隔

アクティビティ概要	
「S3-02-02 目標復旧水準」で定義した RPO(目標復旧地点)、RTO(目標復旧時間)の要件に応じたバックアップ取得間隔を設定する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務ルール定義 ● システム要求一覧 ● システム機能俯瞰図 ● システムフロー ● 外部 IF 機能要件定義 ● バッチ機能一覧 ● 論理データベース要件定義 	<ul style="list-style-type: none"> ● バックアップ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
➤ 障害発生時どの時点まで復旧する必要があるのかにより、データバックアップの頻度を任意、月次、週次、日次、同期といったレベルで設定する。	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-06-05 バックアップ保存期間

アクティビティ概要	
主に可用性の観点で実施されるバックアップの世代管理とは別に、データ保全という観点でバックアップデータの保存期間を設定する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務ルール定義 ● システム要求一覧 ● システム機能俯瞰図 ● システムフロー ● 外部IF 機能要件定義 ● バッチ機能一覧 ● 論理データモデル定義 	<ul style="list-style-type: none"> ● バックアップ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
➤ バックアップデータを障害復旧用途にのみ利用するものとして、データ保存の用途には使用しないことと想定するのか、社内規定や法律で規定されている期間保存する必要があるのかを見極める必要がある。	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-06-06 バックアップ方式

アクティビティ概要	
<p>バックアップ方式を検討し設定する。バックアップにはオフラインバックアップとオンラインバックアップがある。</p> <p>オフラインバックアップとは、システム(あるいはその一部)を停止させてバックアップを行う方式、オンラインバックアップとはシステムを停止せず稼働中の状態でバックアップを行う方式を指す。[2]</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 ● システムフロー ● 外部IF 機能要件定義 ● バッチ機能一覧 ● 論理データモデル定義 	<ul style="list-style-type: none"> ● バックアップ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ リカバリー方式は、「S3-02-02 目標復旧水準」で定めた RPO(目標復旧地点)に対して実現可能な方式になっていなければならない。 ➤ オフラインバックアップはシステム停止を伴うため、可用性を考慮し、バックアップ処理時間とシステム開閉局時間を調整する必要がある。 ➤ オンラインバックアップを行う場合は、バックアップ中の性能目標を考慮する。 	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ DBMS の機能を使う方式やディスクストレージの機能を使用する方式もあるので詳細は方式設計で行うが、ここでは主にオンラインバックアップが必要か否かを確認する。 	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-07 ネットワーク要件の定義

サブプロセス目的
拠点数、接続端末台数、トランザクション量(予想)、プロトコル、トラフィック予想などから各系統の帯域、ネットワーク構成を決定し、ネットワーク構成図を確定する。
サブプロセス概要
拠点業務内容(工場、物流センター等)を踏まえトランザクション量、トラフィック予想から拠点間通信に必要な帯域を確認する。 拠点間ネットワークは必要帯域をもとに障害時のバックアップも含めた最適な WAN サービスを検討する。 構内ではフロア数、フロアごとの端末台数等により LAN 構成を確認する。EDI 用ネットワークは接続先 IP の確認を行う。

S3-07-01 既存回線の利用有無

アクティビティ概要	
<p>現行ネットワーク構成図から既存回線(サービス)の有無を確認する。 新システムにおいて拠点間・拠点内でどのようなデータがどれくらいの量で流れるかを予想し、既存回線(サービス)が新システムに転用可能かを判断する。 既存回線の転用が可能な場合、前提条件や制約条件を定義する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 現行ネットワーク構成図(拠点間・拠点内) ● 現行ソフトウェア構成図 	<ul style="list-style-type: none"> ● ネットワーク要件定義
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 既存回線の拠点間・拠点内の帯域とトランザクションの概略をとらえ現行のトラフィックを予想する。 例えば以下のような観点で新システムのトランザクションと拠点間トラフィックの変化点をまとめる。</p> <ul style="list-style-type: none"> ✓ レガシー画面(固定長データ利用)が Web 画面(HTML 利用)に変更されていないか？ ✓ Terminal Service(画像情報の伝送)の新たな利用はないか？ ✓ 帳票印刷方式の変更によるデータ転送量の変化はないか？ ✓ DWH、BI、ポータル等新規の情報系システムの追加によるトランザクションの増加はないか？ ✓ 切替前の並行運用でレガシープロトコルを残す必要はないか？ <p>➤ 上記のような変化点を基に新システムにおける拠点間・拠点内の必要帯域を概算で導出し、既存回線を利用可否および、利用の前提、制約条件を確認する。</p> <p>➤ 既存回線転用の制約条件として以下を確認する。</p> <ul style="list-style-type: none"> ✓ 既存回線がカバーしているサービスと割当帯域 ✓ 新システムの回線でカバーするサービスごとに割当帯域を管理するのであれば、現行機器が QoS 分離に対応しているのか否か、非対応なのであれば機器交換が必要なのか？ 	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-07-02 WAN 構成

アクティビティ概要	
使用プロトコル、拠点数、拠点別端末数、拠点間トラフィック量等を元に WAN 構成を確認する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 拠点別・アプリ別利用者数 ● 端末台数 ● 現行ネットワーク構成図(拠点間) 	<ul style="list-style-type: none"> ● ネットワーク要件定義
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 例えばクリティカルな業務を行う主要拠点と本社は広域イーサネット又は IP-VPN 等の信頼度・回線速度が共に高い回線を導入し、そうでない拠点はインターネット VPN と回線速度は遅いが信頼性の高いバックアップ回線の組み合わせにするなどの組み合わせを考える事で、業務継続性とコストの双方を加味して決定する。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ 拠点間で利用アプリや人数・台数が異なるので拠点別に必要帯域を出し最適な WAN の構成とする。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-07-03 LAN 構成

アクティビティ概要	
使用プロトコル、拠点内端末数、拠点内トラフィック量等を元に LAN 構成を確認する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 拠点内・アプリ別利用者数 ● 端末台数 	<ul style="list-style-type: none"> ● ネットワーク要件定義
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 無線 LAN を検討する際は有線 LAN、無線 LAN のそれぞれのメリット、デメリットを整理した上で合意する。 ➤ 端末、プリンター、フロア等は今後の増加もある程度見込んでおく。 	
適用技法	
<ul style="list-style-type: none"> ● （特になし） 	

S3-07-04 外部システム接続構成

アクティビティ概要	
外部システム接続の、接続先、接続プロトコル、プロトコル別接続機器等の外部システム接続構成を確認する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● ネットワーク構成図(現行外部企業・団体間連携) ● 外部 IF 機能要件定義 	<ul style="list-style-type: none"> ● ネットワーク要件定義
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 新システム導入にあたり、インターフェースが追加・変更になる、外部システム接続先があれば変化点を洗い出す。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ インターネット利用(流通 BMS)と専用・公衆線利用(全銀・JCA)等の従来型・次世代型のインターフェースが混在する場合で、現行運用に課題がある際はクラウド等の新たな外部サービス利用(*1)も含めて検討する。 (*1) 新たな外部サービス利用に必要なネットワークは「S3-07-02WAN 構成」で検討する。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-07-05 保守回線

アクティビティ概要	
保守作業に必要な保守回線の要件を確認する。 保守を行うに当たりリモートサービスは不要であれば検討の必要はない。	
インプット	アウトプット
● ソフトウェア構成図(保守環境)	● ネットワーク要件定義
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 保守で本番環境での作業が含まれるか否か、含まれる場合はその作業内容とその作業に必要な帯域を確認する。 ➤ お客さまのセキュリティポリシーを確認する。 ➤ 保守ネットワーク構成を設定する際には、端末数、ピーク時流量、拠点冗長性(BCP,複数ベンダ考慮)なども検討する。 	
適用技法	
● (特になし)	

S3-08 移行要件の定義

サブプロセス目的
システム移行に必要な要求項目を漏れなく抽出して、移行方式設計策定へのインプットにする。
サブプロセス概要[2]
<p>移行要件は、現行システム資産の移行に関する要求である。</p> <p>完全に新規で開発する場合を除き、現行システム資産を新システムに移行できなければ、システム開発が完了してもシステムが利用できない。したがって、システム移行に必要な要求事項を漏れなく抽出して移行計画を策定し、確実に実施することが非常に重要である。</p> <p>移行要件は、移行のスケジュールに関する「移行時期」、新システムへの切り替え方式に関する「移行方式」、移行する資産の対象を把握する「移行対象」、および移行のためのリハーサル等を行う「移行計画」から構成される。</p>

S3-08-01 移行時期

アクティビティ概要	
スケジュールの観点で、移行に求められる要求事項を設定する。各要求事項に理由・背景があれば、明確にしておく。	
確認項目	説明
システム移行期間	移行作業計画承認から新システム本稼働までのシステム移行期間。
システム停止可能日時	システム停止可能日時と期間。分断される場合はそれぞれに定義。
新システム本稼働日時	新システムが正常稼働しなければいけない期限日時
並行稼働の有無・期間	並行稼働の有無・期間および対象システムを決定する。
お客さま業務への影響、制約事項	システム移行期間中に、お客さまが被る業務的な影響、制限事項等を明確にする。 合意した制限事項の範囲内で移行計画を策定し、業務影響も合意した範囲に留めることを前提とする。
インプット	アウトプット
<ul style="list-style-type: none"> 業務要求一覧 業務フロー システム要求一覧 システム機能一覧 	<ul style="list-style-type: none"> 移行性要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手に進めるためのポイント、注意事項	
システム移行期間 <hr/> <p>➤ 移行実施期間中に年度・半期・四半期・月が変わる場合は業務上の締めが発生するため、新旧どちらで処理するかなど注意が必要である。</p>	
システム停止可能日時 <hr/> <p>➤ 業務への影響、制約を十分検討して決定する必要がある。 システムによっては、システム停止可能な日や時間帯が連続して確保できない場合がある。 その場合には、システム停止可能日と時間帯をそれぞれ確認する。</p>	
上手に進めるためのポイント、注意事項	
並行稼働の有無・期間 <hr/> <p>➤ 並行稼働は、現行システムと新システムをある一定期間中は両方稼働させ、ユーザーが業務を行う際に新・旧システムを二重運用し、業務品質を保ちつつ新システムの品質を確認するアプローチである。一方で「継続的なデータ移行が必要」「新旧双方の運用コストがかかる」「ユーザーの負担が大きい」など課題も多く、並行稼働採用時は綿密に移行計画を立てること。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-08-02 移行方式

アクティビティ概要	
<p>システムの移行および新規展開時に、「展開先拠点」「展開する業務」の2つの観点で、一斉切り替えを行うか、段階的に切り替えるかを確認する。それぞれに選択理由・背景があれば、明確にしておく。</p>	
確認項目	説明
拠点展開のステップ数	対象システムが、複数の場所に設置されている場合、一斉に展開が可能か、段階的な展開が必要か決定する。
業務展開のステップ数	全業務を一斉切り替えが可能か、業務毎の段階的な展開が必要か決定する。
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>拠点展開の有無・業務展開の有無</p> <hr/> <p>➤ 多段階になるほど新旧両システムの部分的な共存稼働期間が長期化し、共存のための部分データ連携・参照等のしくみや監視要件が複雑化する。したがって、一斉展開より多段階展開の方が難易度は高いため、安易な多段階移行は避け、一括移行・展開時の業務的、システムのリスクを踏まえて判断する。[2]</p>	
上手く進めるためのポイント、注意事項	
<p>業務展開の有無</p> <hr/> <p>➤ 段階的な展開が必要と判断された要因や業務の順序性、システム・データ構造上の依存関係など多角的に展開順序を決定する。</p>	
適用技法	
<ul style="list-style-type: none"> （特になし） 	

S3-08-03 移行対象

アクティビティ概要	
移行前のシステムで使用していた設備において、新システムで新たな設備に入れ替える移行対象設備の内容、移行の必要がある業務データおよびその量（プログラムを含む）を設定する。実現性やコストに対する影響が大きいため、移行対象ごとの移行方法のメドを付けておく。	
確認項目	説明
業務移行	業務移行に関する計画、実施はベンダーの担当外であることが多いが、お客さまが担当することを明確にする。新システム運用開始に伴う、お客さま業務に対して必要対応はお客さま側で対応していただくことを確認する。 (操作マニュアル作成やユーザー教育が、要求事項に含まれる場合は、対応が必要)
設備移行	旧システムで使用していた設備や機器を新システムで使い続けるのか、全く新しい設備や機器に入れ替えるのかなど、入れ替えの範囲を確認しておく。
データ移行	移行対象データとその量、形式、媒体、移行時の加工複雑度、等を明確にする。 アプリケーションプログラムや設定ファイル等の移行要否も確認する。
手順	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務フロー ● システム要求一覧 ● 論理データモデル定義 ● システム機能一覧 	<ul style="list-style-type: none"> ● 移行性要件定義 ● 非機能要求グレード表(確認項目反映済)
【重要】上手く進めるためのポイント、注意事項	
<p>業務移行</p> <ul style="list-style-type: none"> ➤ 業務移行に伴う、様々な業務上の調整、社内周知の必要性をお客さまに認識していただくことが重要である。 ➤ システム移行、データ移行の都合から生じる、業務上の調整事項や制約等は、ベンダーから明示しなくてはならない。 <p>データ移行</p> <ul style="list-style-type: none"> ➤ 現行の状態・内容に責任を持って移行チームに情報提供等を行う担当組織・チームを、移行対象のデータごとに明確化しておく。後続工程で、思い込みや想定を排除した「現物」を拠り所としたデータ移行設計を行うため、移行元データの責任者を把握しておくといよい。 	
上手く進めるためのポイント、注意事項	
<p>業務移行</p> <ul style="list-style-type: none"> ➤ 教育を実施する時期、対象者(役割)を計画としてまとめる。 ➤ 本 FW では、オペレータ教育については「S3-04-04 サポート体制」で規定している。 <p>設備移行</p> <ul style="list-style-type: none"> ➤ 継続利用するハードウェアやソフトウェアについては、保守サポート継続可否や新規設備や機器との互換性を確認する必要がある。[2] <p>データ移行</p> <ul style="list-style-type: none"> ➤ 移行データ量や加工複雑度の把握を怠ると、予定時間内に移行処理が終了しないなどのトラブルを招く。 	

適用技法
<ul style="list-style-type: none">● （特になし）

S3-08-04 移行計画

アクティビティ概要	
移行作業の作業分担、リハーサル戦略とそれに基づく実施回数や範囲・内容、移行中のトラブル時の対応体制や対応プラン等の内容を設定する。[2]	
確認項目	説明
移行作業分担	お客さまとベンダー間の移行作業の役割分担・担当範囲を取り決める。 なお、最終的な移行結果についてはお客さまが確認する。
リハーサル	計画、手順どおりに移行作業を進めれば、正しく移行が完了できるよう、複数回の移行リハーサルを計画する。 各回の目的・対象・確認ポイントを具体的にすること、リハーサル全体での移行品質確保の戦略が明確であること、 異常時の切り戻しもリハーサル対象とすること、が重要である。 指標はリハーサル範囲(主要な正常ケース、全正常ケース、正常ケース＋異常時の切り戻し含む等)、リハーサル環境、 リハーサル回数を使用する。
トラブル対処	トラブル対処とは、移行実施場所に駐在する人員(お客さまと当社の体制、役割分担を含む)や切り戻しタイミングの対応プラン等をあらかじめ取り決めておくことである。
インプット	
<ul style="list-style-type: none"> ● 組織一覧 ● 業務階層定義 ● 業務フロー ● 業務ルール定義 ● システム要求一覧 ● システム機能一覧 	アウトプット
	<ul style="list-style-type: none"> ● 移行性要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>移行作業分担</p> <hr/> <ul style="list-style-type: none"> ➤ 移行作業に関する、お客さまとベンダーの役割・作業分担を明確に規定し、責任の所在を明らかにすることが重要である。以下例。 <ul style="list-style-type: none"> ✓ 旧システムの移行対象データの調査、内容品質保証 ✓ 移行対象データの抽出、転送 ✓ 移行対象データの加工、変換 ✓ 本番システムへの移行対象データ投入 ✓ 本番システムの動作検証 <p>リハーサル</p> <hr/> <ul style="list-style-type: none"> ➤ 移行本番当日の問題発覚は、時間的制約から打てる対策が限られ、最悪の場合切り戻しを行う結果となる。 移行リハーサルでは、検出した問題・課題を確実に潰し、次回リハーサルでその結果、効果を検証して、必要な計画見直しを行うことが重要となる。 ➤ リハーサルでは「手順」と「時間」の妥当性を確認する。 手順は正しいだけでなく、読み手の誤解・勘違い・思い込みによる移行ミスが起きないよう、記述品質にも注意する。 時間は、移行タスクスケジュールとの差異を洗い出し、移行時間の短縮などの検討に繋げる。 ➤ 大規模開発の場合は、短時間で大勢の担当者が移行作業を行うため、情報集約・共有のしくみ、報告ルール、連絡ルートなど、移行本番の運営に着目した確認も必要になる。 <p>上手く進めるためのポイント、注意事項</p> <hr/> <p>リハーサル</p> <hr/> <ul style="list-style-type: none"> ➤ リハーサルでは、正常時、異常時、過負荷時、移行前への切り戻しなどを考慮した内容、確認ポイントを網羅する。 ➤ リハーサル環境要件と、環境構築の役割分担を、明確にする。 	

トラブル対処

- トラブル対処の規定がある場合、その対応体制や対応プランの規定内容について確認する。[2]

適用技法

- （特になし）

S3-09 セキュリティ要件の定義

サブプロセス目的[2]
<p>セキュリティは、構築する情報システムの安全性の確保に関する要求である。</p> <p>構築する情報システムに応じて検討すべきセキュリティに関する非機能要求を明らかにし抜け漏れがないように検討する。</p> <p>適切なセキュリティ対策が講じられないと脅威が現実のものとなり、情報システムを用いた業務の遂行に支障が生じ、その結果として直接的、間接的に大きな経済的、社会的損失をもたらされるおそれがある。</p>
サブプロセス概要
<p>対象システムのセキュリティに関する前提条件・制約条件、開発時や運用時のセキュリティ管理に関する項目である</p> <p>「セキュリティリスク分析」、「セキュリティ診断」、「セキュリティリスク管理」、セキュリティ対策を実機能である「アクセス・利用制限」、「データの秘匿」、「不正追跡・監視」、機能の組合せによるセキュリティ対策の主要なパターンとなる「ネットワーク対策」「マルウェア対策」、「Web 実装対策」等の内容を設定する。[2]</p> <p>セキュリティに関する非機能要求は、情報システムの性能に影響を与える要求が多い。例えば、暗号化処理のように元の情報処理に負荷を加えるような要求が多いためである。セキュリティに関する非機能要求を検討する際には、併せて性能・拡張性に関する非機能要求を検討することが重要である。</p>

S3-09-01 情報セキュリティに関するコンプライアンス

アクティビティ概要[2]	
<p>お客さまが遵守すべき情報セキュリティに関する組織規程やルール、法令、ガイドライン等が存在するかどうかを確認する。 なお、遵守すべき規程等が存在する場合は、規定されている内容と矛盾が生じないよう対策を検討する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 業務ルール定義 ● システム要求一覧 	<ul style="list-style-type: none"> ● セキュリティ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ 確認すべき組織規程、法令、ガイドラインの一例を以下に示す。</p> <ul style="list-style-type: none"> ✓ 情報セキュリティポリシー ✓ 不正アクセス禁止法 ✓ 個人情報保護法 ✓ プライバシーマーク ✓ SOX 法 ✓ ISO/IEC27000 系 ✓ FISC 安全対策基準 ✓ PCIDSS 等 	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-09-02 セキュリティリスク分析

アクティビティ概要[2]	
<p>システム開発を実施する中で、どの範囲で対象システムの脅威を洗い出し、影響の分析を実施するかの方針を設定する。セキュリティに関する非機能要求を合意する際には、セキュリティ対策の実施有無だけでなく、セキュリティ対策を実施しない項目については対策を実施しないことにより残存するセキュリティリスクについても合意する必要がある。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 ● 業務ルール定義 	<ul style="list-style-type: none"> ● セキュリティ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 対象システムにおける脅威の洗い出し、影響分析を実施する適切な範囲を設定するためには、資産の洗い出しやデータのライフサイクルの確認等を行う必要がある。また、洗い出した脅威毎に対策の要否・レベルを検討する。[2] ➤ ネットワークを通じた不特定多数の攻撃者からの脅威にさらされるシステム構成や、重要情報が取り扱われているため脅威が現実のものとなった場合の影響が大きいシステムの場合は、特にシステム全体のリスクを分析する必要がある。[2] 	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● リスク分析 ● CIA(セキュリティ分析のフレームワーク) 	

S3-09-03 セキュリティ診断

アクティビティ概要	
<p>対象システムや、各種ドキュメント(設計書や環境定義書、ソースコードなど)に対して、セキュリティに特化した各種試験や検査の実施の有無を確認する。検討する際は、利用するツールやレビュー方法等、より具体的な診断方法や診断範囲についても検討する必要がある。[2]</p> <p>指標はネットワーク診断実施の有無、Web 診断実施の有無、DB 診断実施の有無を使用する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> セキュリティ要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ ネットワーク診断は、システムに対する広義の診断をあらわしている。ネットワーク診断には、目視による設定の確認や、疑似攻撃を実施することにより脆弱性を発見する診断(ペネトレーションテスト)等がある[2]が、分析された脅威、資産に応じて診断方法および診断の範囲を決定する。外部ネットワークとの接続の有無、インターネット等の不特定多数との接続の有無等の環境に応じた診断、内部ネットワークからも適切なアクセス制限がかけられているか、等の観点が必要である。 ➤ Web 診断とは、Web サイトに対して行う Web サーバや Web アプリケーションに対するセキュリティ診断のことである。[2] 内部ネットワーク上の端末もウィルスに感染し、攻撃元となる可能性があるため、外部ネットワークとの接続ない Web サーバ、Web アプリケーションについても一定のレベルでの診断は必要である。 ➤ DB 診断とは、データベースシステムに対して行うセキュリティ診断のことである。[2] データベースシステムは通信機能を有し、また固有のアクセス制限、認証機能を有している。データベースシステムそのものの特権の管理や脆弱性への対策だけでなく、システム全体の権限管理との整合を保つ必要がある。 	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-09-04 認証機能

アクティビティ概要[2]	
<p>資産を利用する主体(利用者や機器等)を識別するための認証を実施するか、また、どの程度実施するのかを設定する。 多段の認証を実施することにより、攻撃を抑止する効果を得ることができる。 なお、認証方式としては、ID/パスワードによる認証や、ICカード等を用いた認証等がある。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> セキュリティ要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ 攻撃者が管理権限を手に入れることによる、情報漏洩等の脅威に対抗するためには、管理権限を持つ主体に複数回の認証を実行する必要がある。 [2] ※管理権限を持つ主体とは、システムの管理者や業務上の管理者を指す。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-09-05 利用制限

アクティビティ概要	
<p>認証された主体(利用者や機器など)に対して、資産の利用制限のルールを定め、ソフトウェアやハードウェアによって、どのように実現するのかを確認する。</p> <p>例)USB や CD-RW やキーボードなどの入出力デバイスの制限、コマンド実行制限、アプリケーション機能の利用制限など。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> セキュリティ要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 不正なソフトウェアがインストールされる、不要なアクセス経路(ポート等)を利用可能にしている等により、情報漏洩の脅威が現実のものになってしまうため、これらの情報等への不要なアクセス方法を制限する必要がある。(操作を制限することにより利便性や、可用性に影響する可能性がある)[2]</p>	
上手く進めるためのポイント、注意事項	
<p>➤ 利用制限について検討する際には、実施個所(サーバ、ストレージ等)毎に対策の検討を実施する必要がある。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-09-06 データ暗号化

アクティビティ概要	
機密性のあるデータを、伝送時や蓄積時に秘匿するための暗号化を実施するのを確認する。[2] 指標は伝送データの暗号化の有無、蓄積データの暗号化の有無、鍵管理を使用する。	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> セキュリティ要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 秘匿するために暗号化処理を行う場合は性能への影響を考慮する必要がある。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ データ暗号化について検討する際には、暗号化する情報資産および実施箇所について併せて検討する必要がある。[2]</p> <ul style="list-style-type: none"> ✓ 外部ネットワーク経由だけでなく、ローカルネットワーク経由で重要情報を送付する場合も、盗聴等の脅威に対抗するために、伝送データを暗号化する必要がある。 ✓ データベースやバックアップテープ等に格納されている個人情報等やパスワード等の重要情報の漏洩の脅威に対抗するために、蓄積データを暗号化する必要がある。 	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-09-07 不正監視

アクティビティ概要	
<p>不正行為を検知するために、それらの不正について監視する範囲や、監視の記録を保存する量や期間を設定する。[2]</p> <p>取得対象のログは、不正な操作等を検出するための以下のようなものを意味している。</p> <ul style="list-style-type: none"> ✓ ログイン/ログアウト履歴(成功/失敗) ✓ 操作ログ等 	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 	<ul style="list-style-type: none"> ● セキュリティ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ セキュリティログの取得等による性能への影響を考慮する必要がある。[2]</p>	
上手く進めるためのポイント、注意事項	
<p>➤ どのようなログを取得する必要があるかは、実現するシステムやサービスに応じて決定する必要がある。</p> <p>➤ ログを取得する場合には、不正監視対象のサーバ、ストレージ等と併せて、確認する不正の範囲を定める必要がある。</p> <ul style="list-style-type: none"> ✓ 不正監視対象にはサーバ、ストレージ等への不正アクセス、ネットワーク上の不正なパケット、不正な侵入者等がある。 ✓ 脅威が発生した際に、それらの検知と対策を迅速に実施するためには、重要度が高い資産を扱う範囲だけでなく、システム全体を範囲とすることで、侵入経路などの調査に必要な情報を広く保持することができる。 <p>➤ ログの確認間隔は、随時(セキュリティに関するイベントを認識した時)実施でよいのか、定期的を実施するのか、常時実施するのかを設定する。セキュリティに関するイベントとは、重要な脅威が発見された際や、サービス上の不具合等によりインシデントが発生した可能性がある場合を意味する。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-09-08 ネットワーク対策

アクティビティ概要	
<p>ネットワークに関するセキュリティレベルを検討し設定する。不正な通信を遮断するための通信制御を実施するかを確認する。 ネットワーク上における、不正追跡・監視を実施し、システム内の不正行為や、不正通信を検知する範囲を確認する。 また、ネットワークへの攻撃(DoS など)による輻輳についての対策を実施するかを確認する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> セキュリティ要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 通信制御を実現する際には、ファイアウォール等の導入を検討する。[2] ➤ 検知範囲の設定に応じて、IDS 等の導入を検討する。 ➤ 輻輳対策として、サーバ処理能力の増強で対処する場合、性能・拡張性のなかでサーバの処理能力を決めていくリソース拡張性などの項目と併せて検討する。[2] 	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-09-09 マルウェア対策

アクティビティ概要	
マルウェア(ウイルス、ワーム、ボット等)の感染を防止する、マルウェア対策の実施範囲やチェックタイミングを設定する。	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> セキュリティ要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
➤ リアルタイムでの監視を行うことによる性能への影響を考慮する必要がある。[2]	
上手く進めるためのポイント、注意事項	
➤ 実施範囲は重要度の高い資産に限定するのか、システム全体を対象にするのかを検討する。 ➤ 対策を実施する場合には、ウイルス定義ファイルの更新方法やタイミングについても検討し、常に最新の状態となるようにする必要がある。また、マルウェアチェックをリアルタイムで実施するのか、フルスキャンも実施するのかを検討する。 ➤ リアルタイムスキャンを実施する場合は、例えば以下のようなタイミングを検討する必要がある。[2] <ul style="list-style-type: none"> ✓ ファイルサーバへデータをコピーするタイミング ✓ メールサーバがメールを受信したタイミング ✓ ファイルへの入出力処理が実行される前 等 ➤ フルスキャンを実施する場合は、以下のようなチェックタイミングを検討し選択する。[2] <ul style="list-style-type: none"> ✓ 定期 ✓ 1回/月 ✓ 1回/週 ✓ 1回/日 	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

アクティビティ概要	
<p>Web アプリケーション特有の脅威、脆弱性に関する対策を実施するかを設定する。[2]</p> <p>Web システムが攻撃される事例が増加しており、データベース等に格納されている重要情報の漏洩、利用者へのなりすまし等の脅威にさらされている。このため、Web システムを構築する際には、セキュアコーディング、アプリケーションフレームワーク上の設定・対策実装、Web サーバの設定等による対策の実施を検討する。</p> <p>・Web アプリケーション脆弱性の例</p> <ul style="list-style-type: none"> ✓ SQL インジェクション…アプリケーションが想定しない SQL 文を実行させることにより、データベースシステムを不正に操作する。 ✓ CSRF(クロスサイト・リクエスト・フォージェリ)…掲示板や問い合わせフォームなどを処理する Web アプリケーションが、本来拒否すべき他サイトからの不正なリクエストを受信し処理してしまう。 ✓ XSS(クロスサイト・スクリプティング)…XSS 脆弱性のある Web サイトにユーザーを誘導することにより、ユーザー環境で不正スクリプトを実行させる。 	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧 	<ul style="list-style-type: none"> ● セキュリティ要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 対策の有効性を確認するための専門家のレビューやソースコード診断、ツールによるチェック等についても検討する必要がある。[2] ➤ Web 実装対策をすることによる性能への影響を考慮する必要がある。 例えば、WAF(Web Application Firewall)によるリアルタイムでの監視を行うことによる性能への影響など。 	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-10 システム環境

サブプロセス目的[2]
システムの設置環境に関する要求を定義する。 システムを取り巻く利用者や地域的広がり要件は、取り扱うデータ量や設備・ネットワーク構成などコストの影響が大きく、後から変更することが困難である。定義の漏れがあると非機能要件定義全体の手戻りなどが発生し大きなトラブルの要因となりやすいため重要である。
サブプロセス概要
お客さま/ベンダー間で早期にシステムの設置環境に対して共通認識を持っておくことが、システムを構築していくうえで非常に重要である。 例えばユーザー数やクライアント数が正しく定義されていなかったり、システムのライフサイクルの中で将来の増大の考慮が漏れていたりすれば、リソースの問題を引き起こす可能性もある。これらの項目は、早期に合意を図っておくことにより、システムの特性をおさえやすくなる。

S3-10-01 接続対象

アクティビティ概要	
システム利用数、管理対象のクライアント数を確認する。指標はユーザー数、クライアント数を使用する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 組織一覧 ● システム要求一覧 ● システム機能俯瞰図 	<ul style="list-style-type: none"> ● システム環境要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ ユーザー数は性能・拡張性を決めるための前提となる項目であると共に、その規模や地域的広がりによって利用するための端末・ネットワーク等のシステム環境を規定する項目でもある点に留意が必要である。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-10-02 稼働環境

アクティビティ概要	
システムが稼働する拠点の数、国内・国外の拠点展開など地域的な広がりを設定する。 指標は具体的な拠点の所在地、地区・地方ごとの拠点数を使用する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要求一覧 ● 組織一覧 ● システム要求一覧 ● システム機能俯瞰図 	<ul style="list-style-type: none"> ● システム環境要件定義 ● 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
➤ 複数拠点を想定し、拠点数を合意した場合は具体的な値を設定する。	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S3-10-03 特定製品指定有無

アクティビティ概要	
<p>お客さまの指定によるオープンソース製品や第三者製品(ISV/IHV)などの採用の有無を確認する。[2]</p> <p>指定がある場合、製品指定の理由、選定理由、背景、制限事項、適用範囲、ベンダーの責任範囲等を確認する必要がある。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> システム環境要求定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ オープンソース製品や第三者製品(ISV/IHV)の採用によりサポート難易度への影響があるかの視点で確認を行う。[2] <ul style="list-style-type: none"> ✓ ISV: Independent Software Vendor (独立系ソフトウェア ベンダー) ✓ IHV: Independent Hardware Vendor (独立系ハードウェア ベンダー) ➤ ベンダーの責任範囲明確化が重要である。 例えば、導入経験のないパッケージがお客さまから指定された場合、ベンダーが当該パッケージの機能等について調査や検討を行う範囲などに大きく影響する。 	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-10-04 耐震/免震レベル

アクティビティ概要[2]	
<p>地震発生時にシステム設置環境で耐える必要のある実効的な最大震度を設定する。</p> <p>建屋が揺れを減衰するなどの工夫により、外部は震度 7 超でも設置環境では実効的に最大震度 4 程度になる場合には震度 4 よりレベルを設定する。</p> <p>なお、想定以上の揺れではサービスを継続しないでも良い場合には、その想定震度でレベルを設定する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要求一覧 	<ul style="list-style-type: none"> システム環境要件定義 非機能要求グレード表(確認項目反映済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ 設置環境での実効的な震度は、屋外の振動がそのまま伝わる建屋の場合は外部の震度と設置環境の震度はほぼ一致と考えられるので、外部震度からレベルを設定すればよい。</p> <p>ただし、建屋の免震設備などにより、設置環境での最大震度を低く保証できる場合にはその震度を実効的な震度としてレベル設定が可能である。[2]</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S3-11 テスト要件の定義

サブプロセス目的
テスト工程を定義し各テスト工程のテスト内容を決めた上で、工程毎にどのような検証を積み上げて全体品質を確保するかを検討し、お客さまと合意する。
サブプロセス概要
業務要件およびシステム要件をもとに、各テスト工程に関する基本方針、各テスト工程の目的、対象範囲、検証観点、ケース作成基準などを定義する。 各テスト工程における品質リスクを定め品質管理方法を決定する。

S3-11-01 テスト工程定義と各テスト工程の役割分担

アクティビティ概要				
テスト工程定義と役割分担を定義する。				
インプット			アウトプット	
<ul style="list-style-type: none">● 業務要件定義書● システム要求一覧			<ul style="list-style-type: none">● テスト要件定義	
手順				
<div>1. テストの基本的な方針・考え方の定義 各工程のテスト計画書で具体的・詳細な計画を策定できるよう、各工程のテストに含まれる要素(スケジュール・ケース・シナリオ・データ)、環境についての基本的な方針・考え方を定義する。</div> <div>2. テストの配置定義 開発工程の中でどのテストをいつするかを決める。</div> <div>3. テスト対象範囲特定 テスト対象範囲を業務要件定義、システム要件定義から明らかにする。</div> <div>4. テストレベルとテスト種類の選定 どの程度の深さ、抽象レベルで、何に注目してテストするかを明らかにする。</div> <div>5. テスト実施の役割分担定義 テストごとに担当組織とその担当/責任範囲を定義する。</div>				
【重要】上手く進めるためのポイント、注意事項				
➤ 効率的かつ再帰的にテスト実行できるよう、テスト種類ごとにテストツール導入の必要性、可能性を確認する。				
上手く進めるためのポイント、注意事項				
全般				
➤ 機能名・機能ブロック名などを伴う具体的なテストイメージ・スコープについては、後に各テスト計画書で定義する。				
テスト工程定義				
➤ テスト工程定義のサンプルは以下の通り、テスト種類は次項で説明する。				
工程	テスト	テスト概要(目的・対象範囲・検証観点・ケース作成基準)	テスト種類	実施者
単体テスト	単体テスト (JOB/TRAN 単体テスト)	各機能単体の検証を行う。 共通プログラムと結合し、期待する結果が返るか検証する。 画面の場合、存在するリンク・ボタンからの遷移先への検証も含む。 ホワイトボックスで条件網羅のテストを行う。	機能テスト インターフェーステスト	自社
	移行プログラムテスト			自社
	移行 PGM/JOB 単体テスト	各ストアド/JOB 単体の移行プログラムの検証を行う。 移行仕様として問題ないか検証する。 ホワイトボックスで条件網羅のテストを行う。	機能テスト	自社
	移行結合テスト	データ移行プログラム全体がデータ移行設計書を満たしていることを検証する。(組合せ、順序、整合性など)	機能テスト インターフェーステスト	自社
結合テスト	開始判定テスト (全体疎通テスト)	サブシステム内で、正常系処理を中心としたシナリオにより検証を行う。次の2点を確認し、後続のサブシステム連携テストをスムーズに立ち上げることを目的とする。 ① 周辺システムとの IF 整合性を確認する。 ② 本番同様の JOB 設定で正しく稼働する。	インターフェーステスト トランザクションフロー(パス)テスト	自社
	サブシステム連携テスト			自社
	サブシステム内結合テスト	サブシステム内の機能を結合し、機能間インターフェース及びサブシステム内シナリオ検証を行い、画面設計書・帳票設計書通りに実装されていることを検証する。 (サブシステム内の入口から出口までを同一のテストデータを使用し、各機能での整合性がとれていることを確認する。	機能テスト インターフェーステスト	自社
	サブシステム外結合テスト	サブシステムとサブシステムの間、サブシステムとホスト等の連	機能テスト	自社

システムテスト	開始判定テスト (全体疎通テスト)	携を確認する。 システム全体で、正常系処理を中心としたシナリオにより検証を行う。以下を確認し、後続のシナリオテストをスムーズに立ち上げることを目的とする。 ① 周辺システムとの IF の整合性を確認する。 ② 本番同様の JOB 設定で正しく稼働する。	インターフェーステスト インターフェーステスト トランザクションフロー(パス)テスト	自社
	シナリオテスト			
	個別スループットシナリオテスト	システム全体について、業務フローをベースとしたシナリオをもとにサイクル(運用)を考慮したテストを実施することで、システム全体が要件通りに機能・稼働することを確認する。	機能テスト インターフェーステスト トランザクションフロー(パス)テスト	自社
	外部システム連携テスト	関連外部システムと正しく相互接続できることを確認する。 本番と同じ設定・接続を行い、各種パターンで想定している結果が確保されていることを確認する。	機能テスト インターフェーステスト トランザクションフロー(パス)テスト	自社 お客様
	イレギュラーテスト	定期的ではないが、発生しうる業務シナリオの検証を行う。 (組織変更、マスタ変更など)	機能テスト	自社
	インフラ運用性能障害テスト	インフラ環境構築後、インフラ観点での障害テストやマシンスペック通りのスループットが出ているかの性能検証を行う。(障害時のリカバリー運用の検証など)	障害テスト 負荷テスト/ボリュームテスト システム運用テスト	自社
	性能テスト	非機能要件および SLA を満たすことを確認する。 (レスポンス検証。大量データ処理時の性能確認)	負荷テスト/ボリュームテスト	自社
	現新比較テスト	日時、月次処理後の新帳票、旧帳票を比較することにより、現行保証要件を満たすことを確認する。	並行稼働テスト	自社
運用テスト	全体移行リハーサル	本番切替の手順を想定したリハーサルを行う。 主にシステム面に主眼をおいて検証する。 データは運用テスト開始時点のものを使用する。	コンバージョン(移行)テスト	自社
	業務ユーザーテスト	ユーザーが運用テスト計画に基づきシステムが業務要件を満たしているかどうかを検証する。 操作マニュアル等をもとにユーザー視点の使い勝手を検証する。	機能テスト ユーザービリティテスト ユーザー文書及び手続きのテスト	お客様
	現新比較テスト	新処理と現行処理の結果を比較することにより、現行保証要件を満たしていることを確認する。	並行稼働テスト	お客様
	本番移行リハーサル	本番切替を想定した最終的なリハーサルとして、人間系を含めたデータ移行、システム切替、業務移行全般の手順を検証する。	コンバージョン(移行)テスト	自社

- 各テスト工程の役割分担では実施主体が、お客さまかベンダーであるかを確認し、インフラ環境、アプリ環境、テスト仕様書、データの準備についてもお客さまの役割を確認する。

テスト種類

- 「Fintan」が公開している『テスト種別&観点カタログ』が提供する、テストの種別やテストで検証する観点のカタログを活用して、プロジェクトで行うテスト内容を検討するとよい。

テストデータ準備方法

- 各工程のテストデータをどの様に準備するかを決める。例は以下の通り。

テスト工程	準備方法
単体テスト	制約は設けず、各種パターンを網羅できるデータを手動で作成する。
結合テスト	マスタデータは移行プログラムにより本番同様のデータを作成する。 トランザクションデータは移行プログラムで作成せず、システムから生成される想定データを使用する。 ※ベースのデータは仕様に基つき手動作成するが、その後の操作は全てシステム上で行う。
システムテスト	マスタデータは移行 PGM により本番同様のデータを作成する。 トランザクションデータは移行 PGM を使用せず、システムから生成される想定データを使用する。 ※ベースのデータは仕様に基つき手動作成するが、その後の操作は全てシステム上で行う。 ただし、負荷テストのトランザクションデータは、テスト開始時点の移行対象データをベースに、移行プログラムにより作成する。
運用テスト	マスタデータ、トランザクションデータとも移行 PGM により本番同様のデータを作成する。 運用テスト開始時点の移行対象データを全て移行する。

適用技法
<ul style="list-style-type: none">● （特になし）

S3-11-02 各テスト工程の品質管理

アクティビティ概要																																												
<div>各工程テストの品質評価指標は品質マトリクスを定め適用する。</div> <div>これにより、</div> <div><div>✓ 指標による客観的な品質評価</div><div>✓ 品質面でプロジェクト各局面の完了基準の明確化</div><div>✓ 要件、機能、テストの網羅性確保</div></div> <div>を実現することができる。</div>																																												
インプット				アウトプット																																								
● システム要求一覧				● テスト要件定義																																								
手順																																												
—																																												
【重要】上手く進めるためのポイント、注意事項																																												
—																																												
上手く進めるためのポイント、注意事項																																												
全般																																												
<div>➤ 品質マトリクスとあわせてレビュー実施要件を定義し「品質要件」とする場合もある。 この場合は工程毎にお客さまレビュー、内部レビューの対象、実施方法、タイミング、参加者を定義する。</div> <div>➤ テスト結果の記録の取り方、エビデンス保管法(例えばペーパーのエビデンスが必要かどうか、画面キャプチャは電子ファイルで保管可能かなど)を検討する。</div> <div>➤ 不具合の記録方法と不具合記録をどのような手法を使ってまとめて品質評価を行うかを検討する。</div>																																												
品質マトリクスの適用対象と基準																																												
<div>➤ 例は以下の通り。</div> <table><tr><td></td><td>テスト密度</td><td>バグ 検出密度</td><td>ヒット率</td><td>レビュー 工数比率</td><td>レビュー 指摘密度</td><td>カバレッジ</td><td>完了率</td><td>残存問題数</td></tr><tr><td>単体テスト</td><td>100/KLOC</td><td>5/KLOC</td><td>10%</td><td>—</td><td>—</td><td>C2 100%</td><td>100%</td><td>0</td></tr><tr><td>結合テスト</td><td>25/KLOC</td><td>2/KLOC</td><td>7%</td><td>—</td><td>—</td><td>—</td><td>100%</td><td>発生数の2%</td></tr><tr><td>システムテスト</td><td>5/KLOC</td><td>0.5/KLOC</td><td>3%</td><td>—</td><td>—</td><td>—</td><td>100%</td><td>発生数の2%</td></tr></table> <div>各プロジェクトの特性により定める</div>										テスト密度	バグ 検出密度	ヒット率	レビュー 工数比率	レビュー 指摘密度	カバレッジ	完了率	残存問題数	単体テスト	100/KLOC	5/KLOC	10%	—	—	C2 100%	100%	0	結合テスト	25/KLOC	2/KLOC	7%	—	—	—	100%	発生数の2%	システムテスト	5/KLOC	0.5/KLOC	3%	—	—	—	100%	発生数の2%
	テスト密度	バグ 検出密度	ヒット率	レビュー 工数比率	レビュー 指摘密度	カバレッジ	完了率	残存問題数																																				
単体テスト	100/KLOC	5/KLOC	10%	—	—	C2 100%	100%	0																																				
結合テスト	25/KLOC	2/KLOC	7%	—	—	—	100%	発生数の2%																																				
システムテスト	5/KLOC	0.5/KLOC	3%	—	—	—	100%	発生数の2%																																				
適用技法																																												
● (特になし)																																												

S3-12 非機能要件の制約条件の定義

サブプロセス目的
非機能要件の制約条件を定義する。
サブプロセス概要
非機能要件の制約条件には大きく分類すると、ビジネス制約と技術制約がある。 ビジネス制約とは、法令や社内規程、提携先とのインターフェースといったビジネスの前提となるものが該当する。 技術制約とは、適用製品や適用サービスなどが該当する。 非機能要件を定義する上での、これらの制約条件を明確にする。

S3-12-01 非機能要件の制約条件の定義

アクティビティ概要	
<p>非機能要件を定義する上での、制約条件を明確にする。 以下に具体例を示す。</p> <ul style="list-style-type: none"> ✓ 物理的環境の前提・制約条件: 事業所の電源容量、停電用自家発電機有無、耐床持重、空調、免震設備、落雷設備など ✓ 法的前提・制約条件: 参照するセキュリティポリシー 社内基準、J-SOX 法、ISO/IEC27000 系、政府機関の情報セキュリティ対策のための統一基準 プライバシーマーク、FISC、構築実装場所の制限、リモートからの運用の可否など ✓ ネットワークの前提・制約条件: 既存接続先の接続インターフェース、IP アドレスなど 	
インプット	アウトプット
● 非機能要件定義	● 非機能要件定義(制約条件付与済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
—	
適用技法	
● (特になし)	

S3-13 非機能要件の対応レベル決定

サブプロセス目的
これまでに検討してきた個々の非機能要件の対応レベルを、対象システム機能の重要性や品質やコスト等を加味した全体最適な非機能要件の対応レベルにする。
サブプロセス概要
これまでに検討してきた非機能要件の対応レベルについて、全体最適の観点で評価し、必要に応じて対応レベルの補正を行い、対応レベルを決定する。

S3-13-01 対応レベルの補正と決定

アクティビティ概要	
<p>これまでに検討してきた個々の非機能要求の対応レベルを、システム機能の事業における重要性や問題が発生した場合のリスク、システムの開発コストやスケジュール、技術的制約、実装方式のフィジビリティ、非機能要求間の依存関係等から全体最適の観点で評価する。</p> <p>必要な場合は非機能要求グレード表の重要項目を中心にお客さまと一緒に検討し、補正を行い、対応レベルを決定する。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● 非機能要件定義(制約条件付与済) ● 非機能要求グレード表(確認項目反映済) 	<ul style="list-style-type: none"> ● 非機能要件定義(対応レベル確定済)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ 機能要件とのトレードオフも検討する。</p> <p>優先度の高い非機能要件を実現するために弊害となっている機能要件が存在する場合は、その機能要件を取り下げる、もしくは見直すなどの対応が必要になるケースがある。例えば、性能要件の優先度が高い場合、性能に悪影響となる複雑な画面機能要件の取り下げや見直しをするケースなど。そのような場合は、お客さまと優先順位を検討した上で、対応方針を決定する。</p>	
適用技法	
<ul style="list-style-type: none"> ● 非機能要求グレード 	

S4 全体要件の精査

S4-01 要件の検証と妥当性確認

サブプロセス目的
定義した要件の構造や意味が正しいことを確認する。 システム要件定義で定義した要件が、ビジネス要件、業務要件と照らし合わせた際に整合性がとれていることを確認する。
サブプロセス概要
システム要件定義の成果物である画面、帳票、バッチ、外部 IF、論理データモデル、非機能要件定義に定義された内容に矛盾がないことを確認する。 システム要件定義の成果物と、業務要件定義の成果物を比較して、本来の目的との整合性が保たれていることを確認する。

S4-01-01 機能要件と非機能要件の検証

アクティビティ概要	
<p>設定した基準・方法でシステム要件の検証を行う。</p> <p>検証では、機能要件、非機能要件の全体として、要件に矛盾がないこと、漏れなく記述されていることを確認する。</p> <p>※検証の結果、課題が発生した場合は、「S1 システム要求の収集と整理」プロセスに戻る。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム機能一覧 ● システム機能俯瞰図 ● システムフロー ● 画面機能要件定義 ● 帳票機能要件定義 ● 外部IF 機能要件定義 ● バッチ機能要件定義 ● 論理データモデル定義 ● CRUD 図 ● 非機能要件定義(対応レベル確定済) 	<ul style="list-style-type: none"> ● システム機能一覧(全体検証済み) ● システム機能俯瞰図(全体検証済み) ● システムフロー(全体検証済み) ● 画面機能要件定義(全体検証済み) ● 帳票機能要件定義(全体検証済み) ● 外部IF 機能要件定義(全体検証済み) ● バッチ機能要件定義(全体検証済み) ● 論理データモデル定義(全体検証済み) ● CRUD 図(全体検証済み) ● 非機能要件定義(全体検証済み)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>※「要件定義計画プロセスガイド：C2-02-04 検証の基準、方法の設定」の「【重要】上手く進めるためのポイント、注意事項」も参照。</p> <p>※検証方法・観点については、「技法ガイド(検証ガイド編)」で解説予定。</p>	
上手く進めるためのポイント、注意事項	
<p>※「要件定義計画プロセスガイド：C2-02-04 検証の基準、方法の設定」の「上手く進めるためのポイント、注意事項」も参照。</p> <ul style="list-style-type: none"> ➤ 個別に作成している成果物間に不整合が生じることが懸念される。 特定の機能に着目し、複数の成果物を通して一貫した表記がされているかを確認する。 また成果物間の参照関係がある場合、矛盾がないことを確認する。 ➤ 利便性を向上させるための複雑な機能要件と高度な性能要件を併せ持つ機能など、相反する要件が記載されている機能を対象に、実現性、開発やテストの効率・しやすさ、等の観点で妥当性を確認する。 問題がある場合は、機能要件の見直しや、非機能要求のグレードダウン等をお客さまと相談する。 ➤ 機能間の関連、連携の確認のためにシステムフローを参照し検証を行う。 ➤ システム機能が、設計標準や実行処理方式標準等に準拠していることを確認する。 標準に準拠しないことで、「必要な共通部品や方式が提供されずシステム機能の実現が困難になる」「部品共通利用による開発生産性向上メリットを享受できない」などの問題が生じる。 	
適用技法	
<ul style="list-style-type: none"> ● 構造化ワークスルー ● チェックリスト 	

S4-01-02 機能要件と非機能要件の妥当性確認

アクティビティ概要	
<p>設定した基準・方法でシステム要件の妥当性確認を行う。</p> <p>妥当性確認では主に、定義したシステム要件がプロジェクトの目的・目標および業務要件を達成できるかを確認する。</p> <p>※妥当性確認の結果、課題が発生した場合は、「S1 システム要求の収集と整理」プロセスに戻る。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム機能一覧(全体検証済み) ● システム機能俯瞰図(全体検証済み) ● システムフロー(全体検証済み) ● 画面機能要件定義(全体検証済み) ● 帳票機能要件定義(全体検証済み) ● 外部IF 機能要件定義(全体検証済み) ● バッチ機能要件定義(全体検証済み) ● 論理データモデル定義(全体検証済み) ● CRUD 図(全体検証済み) ● 業務要件定義書 ● 非機能要件定義(全体検証済み) 	<ul style="list-style-type: none"> ● 業務要求 vs 機能・非機能要件対応表
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>※「要件定義計画プロセスガイド：C2-02-05 妥当性確認の基準、方法の設定」の「【重要】上手く進めるためのポイント、注意事項」を参照。</p> <p>※妥当性確認方法・観点については、「技法ガイド(妥当性確認ガイド編)」を参照。</p> <p>➤ ビジネス上の価値がある要件であっても、本来の主旨に合致しない要件については、プロジェクトスコープから外すなどを検討する。</p>	
上手く進めるためのポイント、注意事項	
<p>※「要件定義計画プロセスガイド：C2-02-05 妥当性確認の基準、方法の設定」の「上手く進めるためのポイント、注意事項」を参照。</p>	
適用技法	
<ul style="list-style-type: none"> ● ロジックツリー(上位要件と下位要件との関連性を表現するために有効) 	

S5 全体要件の合意と承認

S5-01 要件の実施対象決定

サブプロセス目的
定義されたシステム要件の実施対象をお客さまと合意し、プロジェクトオーナーの承認を得ることで、次工程の実施対象を決定する。
サブプロセス概要
工数、コスト、スケジュールを試算する。その後、ビジネスへの貢献度や、事前に計画した基準に基づいて、実施対象とするシステム要件をお客さまと合意する。合意した結果について、プロジェクトオーナーから承認を得る。

S5-01-01 工数、コスト試算

アクティビティ概要	
定義されたシステム要件を実現するために、次工程以降で必要となる工数、コストを試算する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム機能一覧(全体検証済み) ● 画面機能要件定義(全体検証済み) ● 帳票機能要件定義(全体検証済み) ● 外部 IF 機能要件定義(全体検証済み) ● バッチ機能要件定義(全体検証済み) ● 論理データモデル定義(全体検証済み) ● CRUD 図(全体検証済み) ● 非機能要件定義(全体検証済み) ● 生産性指標などの組織資産 ● 	<ul style="list-style-type: none"> ● 概算見積り
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>全体</p> <hr/> <ul style="list-style-type: none"> ➤ 本アクティビティは、要件定義工程における最終的な工数、コスト試算という意味で定義している。実際には、開発工数・コストの見積と、予算と要求の調整は、計画的に要件定義途上で複数回行うこと。要件定義終了時に一度に調整すると、当初要求から削減後要求のギャップが大きくなり、お客さまとの調整が難航する可能性がある。 ➤ 不確定要素が残存した要件については、見積りの前提事項を明確にし、必要に応じてリスク工数を算出すること。 ➤ 提案時の概算のシステム開発コスト見積もりに対する差異と理由をお客さまに説明し、納得頂くこと。 	
上手く進めるためのポイント、注意事項	
<p>全体</p> <hr/> <ul style="list-style-type: none"> ➤ システム開発総予算と執行時期毎の予算を事前に確認する。 	
<p>算出対象</p> <hr/> <ul style="list-style-type: none"> ➤ 工数、コストの試算単位は、機能毎だけでなく、お客さまが実現対象として選定する機能のまとまりの単位で算出すると、実施対象の選定をスムーズに実施できる。 ➤ 方式設計が最終決定していない場合は、実装方針毎のコストが異なるため、それぞれ工数を算出すること。 	
適用技法	
<ul style="list-style-type: none"> ● 類推法 ● ボトムアップ見積り ● FP 法 ● COCOMO 	

S5-01-02 スケジュール試算

アクティビティ概要	
定義されたシステム要件を実現するために、次工程以降で必要となる開発期間を試算する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 概算見積もり ● システム機能一覧(全体検証済み) ● 画面機能要件定義(全体検証済み) ● 帳票機能要件定義(全体検証済み) ● 外部 IF 機能要件定義(全体検証済み) ● バッチ機能要件定義(全体検証済み) ● 論理データモデル定義(全体検証済み) ● CRUD 図(全体検証済み) ● 非機能要件定義(全体検証済み) ● 生産性指標などの組織資産 	<ul style="list-style-type: none"> ● 概算スケジュール
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 本アクティビティは、最終的なスケジュール試算という意味で定義している。実際には、工数、コスト試算と併せて、スケジュール調整も要件定義中に都度行うこと。すべての要求を受け入れて、要件定義終了時に一気にまとめようとする、当初スケジュールを大幅にオーバーした際に、機能削減などの調整作業が発生し、プロジェクトが進まなくなることがある。</p>	
上手く進めるためのポイント、注意事項	
<p>➤ 本プロジェクトについてのリリース希望日が必達であるのか、及び、必達である理由を事前に確認しておくこと。リリース時期が早まる、もしくは遅くなることで、プロジェクトと各ステークホルダーにどのような影響があるかを事前に確認しておくこと。</p> <p>➤ 各工程の開始時には、前工程の課題整理に時間がとられることが多い。各工程立ち上げ期間を考慮されているかを注意すること。</p> <p>➤ 機能や非機能のリリースを段階的に行おうと考えている場合は、業務要件で定義した業務上のゴールを達成するタイミングとリリースタイミングの整合性がとれている必要がある。</p>	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

S5-01-03 実施対象の決定

アクティビティ概要	
開発コストと導入効果を比較分析し、導入による効果を明確にしたうえで実施対象とする要件をお客さまと合意し、決定する。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 概算見積り ● 概算スケジュール 	<ul style="list-style-type: none"> ● システム要求一覧(決定) ● システム機能一覧(決定) ● システム機能俯瞰図(決定) ● システムフロー(決定) ● 画面機能要件定義(決定) ● 帳票機能要件定義(決定) ● 外部IF機能要件定義(決定) ● バッチ機能要件定義(決定) ● 論理データモデル定義(決定) ● CRUD図(決定) ● 非機能要件定義(決定) ● 概算見積り(決定) ● 概算スケジュール(決定)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>決定時のポイント、注意事項</p> <hr/> <ul style="list-style-type: none"> ➤ 実施対象を絞り込むだけでなく、段階的なリリースにより解決できないかについても検討すること。 ➤ 実施対象とする要件と実施対象としない要件をステークホルダーと合意してエビデンスを残すこと。実施対象としなかった理由や決定経緯を、後工程でもお客さまに説明できるようにしておく。 	
上手く進めるためのポイント、注意事項	
<p>準備でのポイント、注意事項</p> <hr/> <ul style="list-style-type: none"> ➤ 実施対象を選択する単位として、画面機能単位だけでなく業務要件単位ごとにまとめるとお客さま側での検討が容易となる場合がある。 ➤ お客さまに「実施対象を決めてください」、とお願いするだけでは決められない場合がある。いくつかのスコープ案を事前に用意すると良い。 ➤ トップダウンの視点で、要求や機能単位で取捨選択することにより実施対象を絞り込む場合は、「要件定義計画プロセスガイド:C2-02-03 優先順位付け基準・方法の設定」の「手順」や「上手く進めるためのポイント、注意事項」を参照し、準備すること。 ➤ ボトムアップの視点で、要求や機能の見直しにより所要コストを調整する場合は、高度なユーザービリティを求めた複雑な画面を簡易化することや、非機能要求のレベルを見直すことによりコストダウンをはかるなどの準備をすること。 <p>決定時のポイント、注意事項</p> <hr/> <ul style="list-style-type: none"> ➤ 予算や期間の制約で機能削減する場合にも、目的が明確でないと実施対象を判断できない。その為、各システム要件と、業務要件定義で定義したゴールとの紐付けをステークホルダーに再説明し、認識を合わせること。 ➤ 要求が膨らみ、開発対象要件を絞り込む必要がある場合は、要件定義計画で定義した開発可否判断基準や方法をステークホルダーに再説明し、開発可否判断基準の認識を合わせること。 ➤ ベンダーを含めた各ステークホルダー個人の意図によって、恣意的な実施対象選定にならないように注意する。例えば、特定の要求の難易度や複雑さを過大評価して、故意に優先順位を左右しようとするようなことはしてはならない。 	

適用技法

- プライオリティ方式
- 優先順位付けマトリクス
- 100ドルテスト
- イエス／ノー投票
- 4 象限方式(例: 重要度×緊急度)
- MoSCoW(Must、Should、Could、Won't)

S5-02 要件定義書の完成

サブプロセス目的
すべてのシステム要件を正確にかつ漏れなく文書に記述することで、すべてのステークホルダーに理解できるようにする。
サブプロセス概要
作成した成果物を要件定義書としてまとめる。

S5-02-01 要件定義書の完成

アクティビティ概要	
作成した成果物を要件定義書としてまとめる。	
インプット	アウトプット
<ul style="list-style-type: none"> ● システム要求一覧(決定) ● システム機能一覧(決定) ● システム機能俯瞰図(決定) ● システムフロー(決定) ● 画面機能要件定義(決定) ● 帳票機能要件定義(決定) ● 外部IF 機能要件定義(決定) ● バッチ機能要件定義(決定) ● 論理データモデル定義(決定) ● CRUD 図(決定) ● 非機能要件定義(決定) 	<ul style="list-style-type: none"> ● システム要件定義書
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
※要件定義書の作成のポイントについては、「業務要件定義プロセスガイド：G3-01-01 業務要件定義書の作成」の「【重要】上手く進めるためのポイント、注意事項」を参照	
上手く進めるためのポイント、注意事項	
※要件定義書の作成のポイントについては、「業務要件定義プロセスガイド：G3-01-01 業務要件定義書の作成」の「上手く進めるためのポイント、注意事項」を参照。	
適用技法	
<ul style="list-style-type: none"> ● （特になし） 	

S5-03 要件の合意と承認

サブプロセス目的
実施対象となった要件について後続工程以降を開始するために、お客さまと合意し、プロジェクトオーナーの承認を得る。
サブプロセス概要
実施対象となった要件について、システム要件定義書をもとに「C2-03 要件合意と承認ルール」の定義」で定められた方法に従って進める。 後続工程以降の対象となる要件内容をお客さまと合意し、プロジェクトオーナーから承認を得る。

S5-03-01 お客さまによる要件の合意と承認

アクティビティ概要	
<p>実施対象となった要件について、システム要件定義書に漏れなく記述されていることをお客さまに確認していただく。 事前に定められた方法と合意を担当されるお客さまとの間で、基本設計の対象とすることを合意し、プロジェクトオーナーから承認を得る。</p>	
インプット	アウトプット
<ul style="list-style-type: none"> システム要件定義書 	<ul style="list-style-type: none"> システム要件定義書(承認済み) 合意記録 承認記録
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
—	
上手く進めるためのポイント、注意事項	
<p>➤ 要件定義計画プロセスで定義した合意と承認ルールをステークホルダーに再認識して頂いてから進めること。</p>	
適用技法	
<ul style="list-style-type: none"> (特になし) 	

S6 引継ぎ

S6-01 設計工程への引継ぎ

サブプロセス目的
システム要件定義で解決していない事項や、システム要件定義で、設計工程以降で定義すべき事項について、設計工程の担当者と共有を行う。設計工程以降に影響を与える情報が伝わらないことによる、設計作業の抜け漏れや手戻りを防止する。
サブプロセス概要
システム要件定義で解決していない課題やシステム要件定義で導出された設計時に定義すべき事項等、設計工程の開始に必要な情報を整理する。その上で、設計工程の責任者に引継ぎを行う。

S6-01-01 設計工程への申し送り事項整理

アクティビティ概要	
システム要件定義で解決していない課題や、システム要件定義で導出された設計時に定義すべき事項等、設計工程の開始に必要な情報を整理する。	
インプット	アウトプット
<ul style="list-style-type: none"> システム要件定義書（承認済み） 	<ul style="list-style-type: none"> 申し送り事項一覧
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<p>➤ 申し送り事項は、以下の情報も合わせて整理すること。 優先度、重要度、緊急度、影響範囲・内容、担当者、期限、リスク</p>	
上手く進めるためのポイント、注意事項	
—	
適用技法	
<ul style="list-style-type: none"> （特になし） 	

S6-01-02 設計工程への引継ぎ

アクティビティ概要	
システム要件定義と、設計工程以降では、責任者や担当者が異なる場合がある為、システム要件定義で解決していない事項や、システム要件定義で、設計以降で定義すべき事項があった場合など、設計工程の担当者で共有を行う。	
インプット	アウトプット
<ul style="list-style-type: none"> ● 業務要件定義書(承認済み) ● システム要件定義書(承認済み) ● 申し送り事項一覧 	<ul style="list-style-type: none"> ● (特になし)
手順	
—	
【重要】上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ 設計工程への申し送り事項(残課題)による影響について、設計工程の責任者と認識を合わせる。申し送り事項が残っている場合に、次工程の実行自体が不可能となるようなものがないかを、次工程の責任者に確認してもらうこと。次工程の責任者に申し送り事項を受け入れてもらうために、現工程の責任者が実施しなければならないことがないかを明確にすること。 ➤ 取り下げになった要件についても、除外理由や決定経緯の引継ぎを行う。設計工程以降で、再度検討対象に挙がるケースや要件定義工程に関与していなかったお客さまに対して除外した経緯などの説明が必要になるケースがあるため。 	
上手く進めるためのポイント、注意事項	
<ul style="list-style-type: none"> ➤ プロジェクトメンバー間での共有を容易にするため、経緯や留意点は参照しやすい形で残すこと。 ➤ キーマンの入れ替えがある際は、本サブプロセスを参考に十分な引き継ぎを行うこと。お客さま側の体制が変わる場合にも、新たに加わる人への引継ぎを実施してもらうようお願いする。 	
適用技法	
<ul style="list-style-type: none"> ● (特になし) 	

参考文献

- [1] IPA/SEC（2010）『機能要件の合意形成ガイド』

<http://www.ipa.go.jp/sec/softwareengineering/reports/20100331.html>

- [2] IPA/SEC（2010）『非機能要求グレード』

<https://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html>