

付録1. 予兆事例集

	予兆, 失敗	時期	状況	状況に至った理由	問題点
事例1	「そんな説明じゃ伝わらない！」とメンバーに言われた	機能仕様の説明レビュー	・外部委託するシステムの説明で機能仕様書を読んだだけだった。	担当になった項目に関連する機能を簡潔に述べようとした	・開発するソフトウェアのユーザーが誰なのか理解していない ・同様にその目的, 利用状況, 役割, 価値など, 背景を理解できていない
事例2	UIに関してこうしたい方が良いのでは? と提案した際に「それは使わないよ」と言われた	機能検討	・UIに関してこうしたい方が良いのでは? と提案しても, それは使わない, 使いにくくなる, と指摘を受けてしまう。	機能を検討している際によりアイデアであると考えた	・開発しているソフトウェアの内容が理解できていない, 要求を理解していない ・開発するソフトウェアのユーザーが誰なのか理解していない ・同様にその目的, 利用状況, 役割, 価値など, 背景を理解できていない
事例3	ユーザーから特に反応が無い 「もう少し待ってください。そうしたら見せます。」と開発者が言っている	開発途中	・あるシステムにおいて, 計算式の微妙な間違い, 及びテスト不足の原因による不具合がユーザーテスト段階で発覚してしまった。	仕様間違いは無く実装に集中しようとしていた	・アウトプットからのフィードバックを買っていない ・開発側からのアウトプットが少ない
事例4	実際にソフトウェアが動作する環境と, プロトタイプ及び設計で考えている環境が乖離し始めている	要件定義	・データが1000件を超えると, 3秒以内に画面を表示するのが難しくなる。表示項目をページに分割する事で速度は改善されるが, ユーザーは1ページで一度に見たい要望が強い。仕様検討時にはデータの送料が確定していないため検討できなかった。 ・帳票の入力値妥当性を検証する際に, エラーが一つ一つ段階的に出てくるのが煩雑に感じた。そこで一度に出すようにしてみた。しかし最近は入力すると同時にValidationを検証して横にエラーメッセージが出るものもある。UX的には後に述べたものになれるほど優れていそうであるが, システムのパフォーマンスに影響が出やすい。パフォーマンスとUX要素のどちらを優先すればよいのか? ・ソフトウェアの構造を理解していないUIデザイナーが先走りしてしまった。	ユーザーの目線からはすぐれた仕様であるが, ソフトウェアの構成, 期間を考えると出来ないことがある	・開発における不確定な要素を何時まで確定するかを検討できていない ・UIデザイナーとソフトウェア設計者との間のコミュニケーションが不足している
事例5	要件定義時にUIに関する議論を行わない	要件定義	・要件定義の際に機能・データ項目の過不足が主な議論となり, 画面に収まらなかった場合の対応などは話し合われない。	機能, データ項目がそろえばユーザーは満足すると考えていた	・最終的にユーザーがどのようにソフトウェアを利用するのか具体的な考えがない
事例6	そこはかたない違和感	ユーザーテスト	・「これでは説明が不足ではないか」がユーザーテスト時に課題となり, 線, 色, 注意書きの追加が行われ, シンプルなデザインが損なわれる。「画面がさびしい」が課題となり対応が発生する。	説明が無くても開発者の意図はユーザーに伝わっていると考えていた	・ユーザーからのフィードバックが不足している
事例7	決める意志がないミーティング	各ミーティング	・Flashがそろそろ世に広まろうとしていた頃, 社内で技術検討会が開かれた。Java等との比較表を求められ作成する。表現力や開発難易度などの項目に点数を付けて説明をするが, 評価に個人差があるよねと建設的議論の場にならない。	表でまとめて比較すれば解決すると思っていた	解決すべき問題に目が向いていない。何をを使うかではなくて, 何を達成したいのかという軸がなければ, 机上の論争になってしまう

付録1. 予兆事例集

事例8	「超特急」と言われた	要件フェーズ 序盤 要件フェーズ 中盤 実装～検証フェーズ	・プロダクトオーナーが「時間がない」や「急いで」しか言われない。提案をしても「じゃあ、それで」といわれる。	時間がないにとしては決まっていないことが多い	<ul style="list-style-type: none"> ・解決すべき問題に目が向いていない ・どんな問題を解決したいのかが聞けないどんな問題を解決したいのかが聞けない ・真剣味が伝わってこない ・アイデアがどれ程有効かの検証が聞けない
事例9	プロダクトオーナーが目的が明確ではなく、意図が共有できない要件をあげてきた	要件定義	<ul style="list-style-type: none"> ・プロダクトオーナーのあげた要件が何を目的としたのか本人も説明できない。 その要件は他の要件とは目的が異なるものであった。しかも優先順位が高い。 プロダクトオーナーの要件目的を明確にすることをせず、要件を満たすためのスケジュール、タスク洗い出しを実施した。 プロジェクト内では、意見が非常に拡散していった。皆、何を検討、設計すればよいのだろうかと考えていた。 	プロダクトオーナーの強い希望により要件を実現するための手順を踏んだ	<ul style="list-style-type: none"> ・利用者が明確になっていない ・ソフトウェアを開発することでどのような問題が解決されるのか理解していない
事例10	ユーザーの部門間で連携がとれていない雰囲気を感じられる	要件定義	<ul style="list-style-type: none"> ・ユーザー受け入れテストの段階でシステム利用想定部門が不足していることが発覚した。 当初想定した部門だけではなく、データを分析することにより他部門も利用したかった。 ソフトウェアでは閲覧できるデータ範囲を当初想定していた部門に限定し、他部門からはデータは閲覧できないように制限していた。結果、データを分析する部門が利用できないシステムが完成してしまった。 ユーザーの他部門において機能の定義が異なっていた。両部門とも利用するため、定義の相違は軽微だったがシステム改修範囲は拡大した。 ユーザーとの打ち合わせ時に他部門の関係者は出席した段階で発覚。 ・発注者と、実際にシステムを使う現場の間にすでに乖離がある。 	ユーザーの部門間の関係にまで立ち入ることは出来ない判断した	<ul style="list-style-type: none"> ・システムの利用者が誰なのか明確になっていない
事例11	ユーザー利用時に不満の声を上げる	ユーザートレーニング、ユーザー利用	<ul style="list-style-type: none"> ・ユーザートレーニング中に「出力した帳票が分かりにくい」「本当に採算や売上が網羅的に確認できるのか不安」という意見が上がった。 ・帳票レイアウトをExcelで作成して、業務部門との大筋の仕様合意を行なった。その後、帳票ツールで作成したサンプルで再度業務部門と認識合わせを行なった。しかし、PDFで出力すると問題なさそうだが、プリンタに印刷すると印字文字が太くなって見づらく、帳票が分かりにくいことが分かった。大きく2回の打合せをしたが、リリース直前で上記記載の意見が上がった。 ・エラーメッセージの内容がユーザーに伝わっていない。 ・思っていたより表示も文字が多く文字が見切れてしまう。プロトタイプ時には見逃していた。 	プロトタイプ、打ち合わせは数回行っているが見落とししている箇所があった	<ul style="list-style-type: none"> ・最終的なアウトプットをユーザーと共有していない ・ユーザーの実際の利用環境に則したテストを実施していない