

74

顧客の運用観点を取り入れたシナリオテスト抽出方法と その適用効果¹

1. 概要

アンリツネットワークス株式会社（以下、アンリツネットワークス）は通信インフラ市場をターゲットとした WAN 高速化装置や帯域制御装置などのネットワーク装置や遠隔監視制御装置などを開発・販売している。これらの製品は主に組み込み開発のソフトウェアを搭載しており、ソフトウェア開発プロセスモデルはウォーターフォールモデルに従っている。また、派生開発が多いことも特徴のひとつである（図 74-1）。

アンリツネットワークス 開発製品特徴

ネットワーク機器, 監視応用システムの
設計/製造/販売/構築/保守

『つなぐ』
Being Smart & Skillful

- 開発製品/市場
 - WAN高速化装置, 帯域制御装置
 - 遠隔監視制御装置, アナログIPコンバータ
 - 社会インフラ/通信インフラ
- 開発の特徴
 - 組み込み系ソフトウェア
 - ウォーターフォールモデル
 - 派生開発が多い



図 74-1 アンリツネットワークス開発製品の特徴

近年、社会生活におけるネットワークへの依存は高まり、その結果としてネットワークの重要性も年々高まってきている。アンリツネットワークスの製品は社会インフラ市場への納入が多く、顧客の運用を止めることのない製品品質を求められている。

そのような社会的背景や市場要求を満足する製品品質を担保するため、会社全体でソフトウェア開発プロセスの改善活動に取り組んでいる。その改善活動の一つとして、開発プロセスにおける妥当性確認を従来の仕様書ベースの機能を中心としたテストから、顧客の視点に立った製品のライフサイクルに沿って運用条件を抽出したシナリオテストに変更した。

本編では、そのシナリオテスト項目策定の工夫やテスト実施効果について事例を紹介する。

¹ 事例提供: アンリツネットワークス株式会社 森中 秀明 氏、関水 宗樹 氏

2. 状況分析と解決すべき課題

2.1. 状況分析

アンリツネットワークスのソフトウェア開発プロセスはウォーターフォールモデルに沿って実施されており、①要求定義②各種仕様書作成③コーディング④単体テスト⑤結合テスト⑥機能テスト（検証）⑦妥当性確認のプロセスで開発を行っている。特徴的なところは、検証（Verification）は開発部門が実施しているのに対し、妥当性確認（Validation）は品質保証部門が実施している2つのV字モデルとなっている点である（図74-2）。

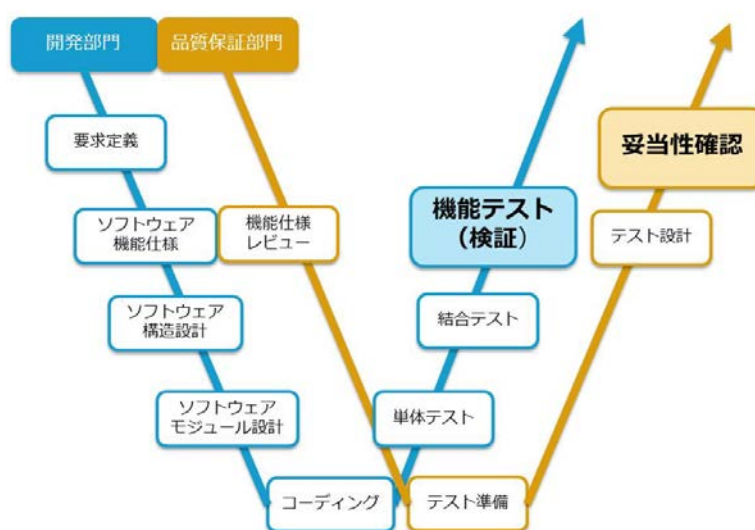


図 74-2 V字モデル 開発プロセス

しかし、検証および妥当性確認の両テスト工程をそれぞれ独立した部門で実施・合格して出荷したにもかかわらず、顧客先で運用を開始するとソフトウェア起因の不具合が発生してしまっていた。そこで、過去に流出した不具合の1つ1つについて、V字モデル左側の「品質を作り込む工程（上流工程）」で作り込んでしまった原因と、V字モデル右側の「品質を確認する工程（下流工程）」での流出原因を分析した。その結果の抜粋を表74-1に示す。

表 74-1 流出した不具合事例

	不具合事例	品質を作り込む工程における 作り込み原因	品質を確認する工程におけ る流出原因
①	回線断の通信異常を検出しない	監視フラグ処理を変更した際回線監視が解除されない	回線断時の 複合条件のテスト不足
②	save API の動作不具合	別タスクで同一メモリを参照しメモリ競合が発生	API コールと他動作の 複合条件のテスト不足
③	親機 LCD に子機情報の一部が表示されない	子機に特定のユニット実装時の仕様定義不足	親機構成の テスト構成の網羅不足
④	リモコンが効かなくなる	装置時刻を 5 分以上進めるとリモコン制御を解除していない	時刻変更テストの リモコン動作確認抜け
⑤	パケット損失発生後通信異常が継続する	パケット損失により送受信処理が重なり CPU 負荷増	異常ケースの 条件検討不足

作り込み原因は、仕様定義段階での検討不足や設計段階での排他処理ミスなど、上流工程プロセスでの様々な要因がある一方、下流工程プロセスで流出した原因については、テスト条件の漏れやテストケースの検討不足によるものとわかった。

2.2. 解決すべき課題とアプローチ方法の検討

下流工程のプロセスにおいて不具合流出した原因を分析した結果として、テスト条件が不足しているという課題が浮かび上がった。この課題解決に向けた対策として取られる代表的な対策は以下がある。

1つは上流工程への対策として、上流工程の仕様定義や設計プロセスの品質を上げることによって、下流工程である検証プロセスと妥当性確認プロセスへ潜在流出する不具合数を減らし、全体として出荷する製品に対する流出する不具合を減らす方法である。もう1つは下流工程への対策として、テスト条件が不足しているという観点から、検証プロセスと妥当性確認プロセスのテスト網羅性を高めることにより流出する不具合を減らす方法である。

前者の対策は一般的に、上流工程のプロセス改善が品質・費用・納期面で最も効果的と言われており有効性が高い。しかし本編では上流工程プロセスの改善とは別に、下流工程のプロセス改善へフォーカスするアプローチを試みたため採択から外した。

後者の対策は、テスト網羅性を上げるために単にテスト項目を増やすだけでは費用面や納期面へ大きく影響が出るおそれがあった。そこで下流工程のプロセスである検証プロセスと妥当性確認プロセスを不具合がすり抜ける要因について掘り下げて調査を行った。V字モデルにおいて、検証プロセスはソフトウェア機能仕様に対応してテスト実施する。一方、妥当性確認プロセスは要求仕様に対応したテスト工程である。アンリツネットワークスの検証・妥当性確認プロセスはそれぞれ要求仕様、機能仕様をインプットとしてテスト項目定義していたものの、個々の仕様、機能を確認することが主目的となっており、テスト観点が似通っていた。

顧客運用下で発生するイベントや利用条件の詳細までは仕様書に記載されていないこともあり、それぞれのプロセスで同じ様なテスト条件不備のテストケースを定義してテスト実施している状況となり、どちらのテストでもテスト条件の漏れが生じていることがわかった。(図 74-3)

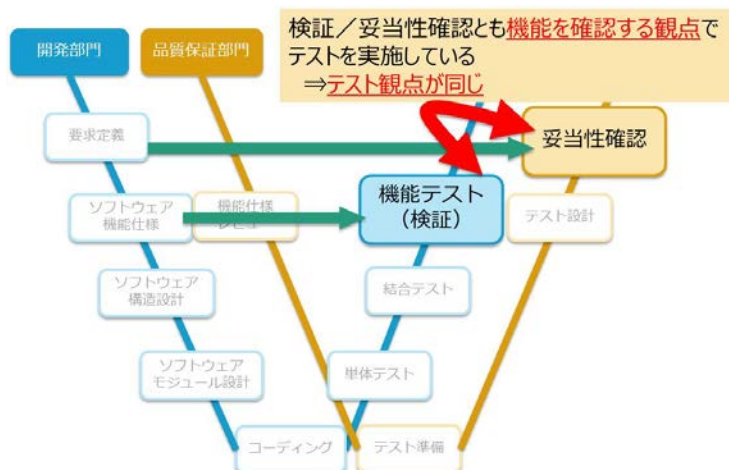


図 74-3 開発プロセスの問題点

そこで本編では検証プロセスと妥当性確認プロセスのテスト観点を異なる観点に変えることで、テスト網羅性を高める取り組みを行った。

検証プロセスは従来どおり、仕様書ベースで正しく作られていることを確認する。回帰テストを実施することで機能仕様の網羅確認を行う。

一方、妥当性確認プロセスは、要求定義通りに作られていることを確認する設計の妥当性確認を行うというテストアプローチをやめ、従来とは異なるテスト観点でテストを実施し、製品としての妥当性確認をすることで不具合の流出防止を図ることとした(図 74-4)。目標は、テスト品質を落とすことなく、テストコストを従来テスト手法と大きく乖離しないで、前年度の流出不具合数より減らすこととした。

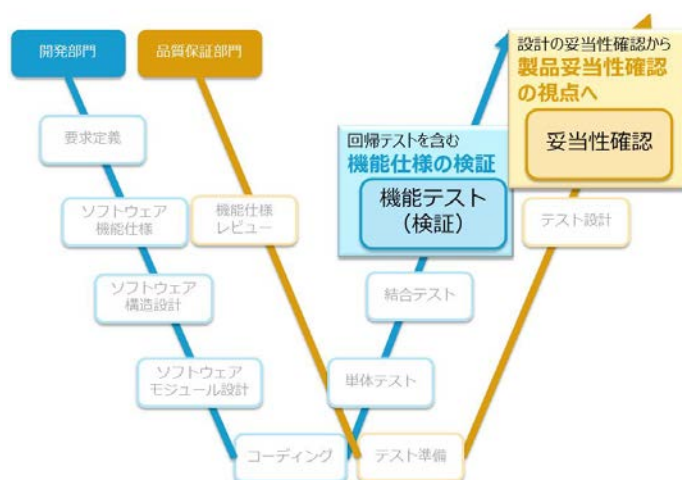


図 74-4 開発プロセスにおける妥当性確認の変更

3. 実施内容

3.1. テスト基本方針の確認

顧客運用条件の定義にあたって、まず基本的な方針を考え、妥当性確認においては顧客要求が反映されていることを確認することが最重要の基本方針であることとした。次にその要求を具体的な項目として落とし込む際に考えたのが ISO 9000:2015²⁾に書かれている、顧客の意図する用途または適用、つまり、顧客の運用状況で望む動作することが重要と考えた（図 74-5）。そこから、我々は顧客の視点に立った運用ライフサイクルを定義し、その運用ライフサイクルにおいて想定される利用方法を確認することをテストの基本的方針と定めた。

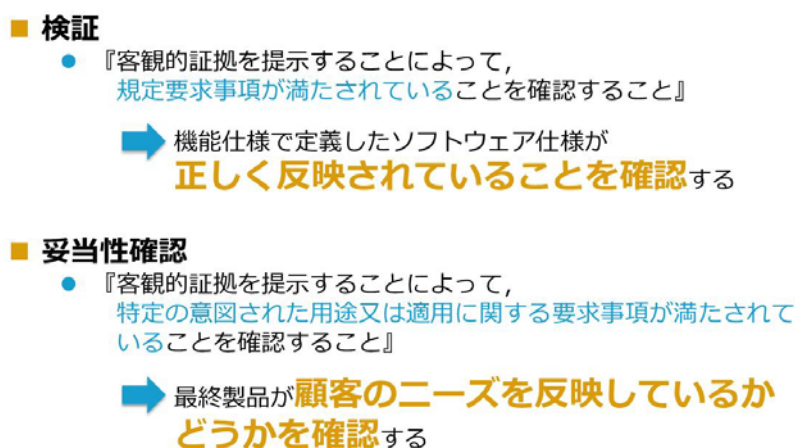


図 74-5 検証と妥当性確認の基本方針

3.2. 運用ライフサイクルの定義

顧客の利用シーンを想定し、製品出荷から設置や保守など製品が実際に市場で発生する条件を時系列的に考えることで、運用ライフサイクルを7項目に分類定義した。

- ① 工場出荷に関する項目
社外へ製品出荷するフェーズの作業を定義する。たとえば、初期値設定、アクティベーション作業など。
- ② セットアップ・設置に関する項目
顧客先で製品を受領し、装置動作を決めるコンフィグレーションを作成、設定し、当該製品を使用可能にするフェーズの作業。たとえば、設置先でのコンフィギュレーション投入や外部機器との接続手順など。
- ③ 定常運用の項目
安定した運用に入ったフェーズでの作業。基本的には安定した連続稼動状態を想定したもので、代表的な機器構成での通信試験や連続運転試験などでのリソース変化確認。

²⁾ ISO 9000:2015（JIS Q 9000:2015）

- ④ 日常業務に関する項目
 ③の定常運用を継続している前提で、顧客が日常行う監視業務や設定変更などに特化して想定したテストなど。たとえばログイン作業、ログ確認、リソース監視など。
- ⑤ 外部環境を変化させる項目
 ③の定常運用を継続している前提で、当該製品・システムの外部事象が変化する場合。たとえば停電／復電による復旧動作や装置への異常パケット入力など。
- ⑥ 保守メンテナンスに関する項目
 ③の定常運用を継続している前提で、保守作業を行う場合。たとえば故障交換、保守員による点検作業、ソフトウェアバージョンアップ作業など。
- ⑦ ユーザビリティに関する項目
 ③の定常運用を継続している前提で、UI/UX といったユーザビリティや感覚的な判断を行う場合。たとえばメニュー構造の連想のしやすさ、GUI の色使い、操作のしやすさや間違いにくさなど。

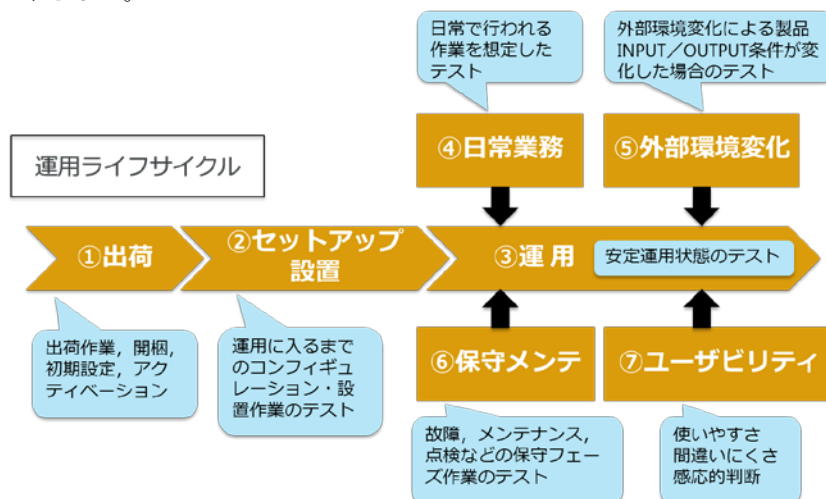


図 74-6 運用ライフサイクルの項目概要

3.3. シナリオテストケースの作成

定義した7つの運用ライフサイクルにおいて、そのサイクルで考えられる顧客の利用シーンや使い方を抽出してシナリオテストケースとして定義した。これが妥当性確認における1つ1つのテスト項目である。

このシナリオテストケースの定義の際には、①過去に流出した不具合事例を反映、②運用オペレーションを顧客からヒアリング、③部門外の有識者とレビュー、の3点の取り組みを実施することによりテストケースの網羅性を高めた。それぞれの内容について下記に説明する。

① 部門外の有識者とのレビュー実施

運用ライフサイクルとして定義した「工場出荷」や「保守メンテナンス」で行われる作業は、製品に関連する社内のサービス部門や出荷責任部門に豊富な知見を有する。そこでこれらに関連する分野について深い知識を持つ参加者を選んで、シナリオテストケースのレビュー

を実施した。また製品や対象市場にわたって広く深い知見を持つ有識者にもレビューに参加してもらうことで、シナリオテストの網羅性を上げる活動を行った。

② 運用オペレーションを顧客からヒアリング

装置の実際の使われ方は製品を購入して運用している顧客に聞くのが最も正確である。そこでステークホルダーであるお客様や販売パートナー、システムインテグレーター等へテスト設計担当が直接足を運び、製品品質を高める目的を伝えた上で、先に定義した7つの運用ライフサイクルにおける実際の運用構成やトラフィック種類、実施されているオペレーションや関連手順など多岐にわたるヒアリングを行った。どのお客様も好意的に協力してくださった。その効果として、シナリオテストケースの不足分を埋めるだけでなく、自分達が想定した顧客先で使用されていると思っていたテスト条件が、実はユーザの利用条件には適合していなく、テスト条件やテストケースとしては優先度が低くしてもよいことが分かる効果もあった。これらのステークホルダーからの意見は、テスト網羅性を高める貴重な意見収集となった。

③ 過去に流出した不具合事例を反映

品質保証部門では、過去に流出した不具合事例とその再発防止策をまとめており、妥当性確認のテスト項目作成のチェックリストとして使うための評価指針ドキュメントを作成していた。

シナリオテストケースを作成する際にも、この評価指針ドキュメントを参照し、過去流出してしまったテスト不足原因は何だったのか、想定していなかった顧客の使用条件は何だったのかを抽出し、今後の流出再発を防止するためにシナリオテストケースに加えた。

品質保証部門でのシナリオテストケース立案と、上記3つの取り組みを行うことによって、図74-7のシナリオテストケースを作成した（一部抜粋）。下記に一部の例を示す。

- 工場出荷の項目では、新規に製造した状態と同じ初期状態でテスト開始するようにした。
- セットアップ・設置の項目では、装置の電源投入と装置設定のタイミングと周辺機器の電源投入、装置との接続タイミングなどの網羅的なテストを追加した。
- 日常業務の項目では、顧客で使われているアプリケーションを使用するようにした。
- 過去に流出した不具合事例の反映では、フラグメントしたIPフレームの順序が入れ替わると転送遅延が発生する不具合の再発防止として、逆順にフラグメントしたIPフレームをトラフィック入力するテストを追加した。

項番	大項目	目的	項番	シナリオ項目
1	工場出荷状態	装置条件として、製造後出荷と同じ初期状態とする		起動 ライセンス設定作業 立ち上がり時間（初期状態，一般的，最大） ...
2	セットアップ・設置	運用開始の条件 装置状態と周辺条件の組み合わせなど	1	運用構成パターン 運用に入るまでの設定過程 設定→接続 接続→設定 ...
			2	各種外部機器・ケーブルの互換性確認
			1	UTPケーブル
			1	ケーブル長（100m～10m）
			2	cat5, cat5e, cat6
			3	ストレート/クロス
			2	光ファイバケーブル
			1	ケーブル長（1m～数km）
			2	SMF/MMF
			3	...
			3	ケーブルの接続間違い
			1	RJ45ネットワークポートとシリアルポート
			2	光ファイバのTxとRx
			3	4W結線ミス
			4	...

図 74-7 シナリオテストケースの内容

3.4. シナリオテストケース作成における工夫点

本編のシナリオテストケース定義の際の工夫点を以下に示す。

① シナリオ想定ポイントの記載

テスト設計者が当該シナリオテストケースをどのような意図や狙いで作成したのかシナリオ想定ポイントを、テストシナリオ項目の横に記述する欄を設けた（図 74-8）。

このシナリオ想定ポイントを記述することで、テスト設計者とテスト実施者の間でテストの意図が共有しやすくなること、レビューをする際に、レビューに対してシナリオテスト項目の作成意図が伝わりやすくなり、効率的なテストレビューができるなどのメリットが生じた。

項目	シナリオ項目	シナリオ想定ポイント
1	各種外部機器・ケーブルの互換性確認	ケーブルや外部機器は様々なスペックがある。どれがきて大丈夫なことを想定
1	UTPケーブル	UTPケーブルを使って接続する機器の場合
1	ケーブル長 (10m~100m)	ケーブル長10m~最長100mまでのケーブル使ったことを想定
2	cat5, cat5e, cat6	ケーブルカテゴリ-cat5, cat5e, cat6, どれが使われるか
3	ストレート/クロス	auto-MDIXが有効かとしてストレート/クロスどちらも使えるか
2	光ファイバケーブル	光ファイバケーブルを使って接続する機器の場合
1	ケーブル長 (1m~数km)	ケーブル長は1m程度の短いものもあれば、数kmの長いものもある ファイバのモードは2種類あってどちらも使う
ポイント <ul style="list-style-type: none"> ● テスト設計者/テスト実施者間で利用シーンの背景を共有できるようにした ● テストレビュー効果効率化 		
3	操作ミス	オペレーションミスを想定する
1	ログインパスワード間違い	Serial/Telnet/WebGUIなどのログイン/admin/パスワードを間違えている
2	Telnet/SSHポート間違い	Telnet/SSHのポート番号を間違えている
3	SNMPコミュニティ名間違い	SNMPのコミュニティ名を間違えている

図 74-8 シナリオテストケースのシナリオ想定ポイント

② テスト条件漏れの防止

顧客利用のテスト観点でテストケースを定義すると、運用構成や機能といった要素がテスト目的そのものではなくテスト条件の一部となる。そのため、シナリオテストケースによるテスト定義では、これらがテスト条件漏れとして表面化する場合がある。そこで、シナリオテスト項目に対して、使用する運用構成や機能の組み合わせをマトリクス形式で関連付けて可視化することで運用構成や機能のテスト条件漏れの防止を図った（図 74-9）。

項目	大項目	目的	項目	シナリオ項目	構成指示	機能網羅
5	運用中の外部環境変化	外部環境変化による装置INPUT/OUTPUT条件が変化した場合のテスト	1	停電・復電	○	○
			2	自装置	○	○
			3	対向装置	○	○
			4	ラック単位	○	○
			5	機器室単位	○	○
			6	建屋単位	○	○
			7	地域一帯	○	○
			8	オペレーション中の電源復旧補償	○	○
			1	上位L2変更	○	○
			2	上位L3変更	○	○
			...			

構成・使用する機能の関連付け

ポイント
テスト構成や機能との**組み合わせ漏れ防止**

図 74-9 シナリオテストケースと機能のマトリクス

4. 適用結果、効果分析

4.1. メトリクス比較による効果確認

運用ライフサイクルに基づいて作成したシナリオテストと過去に行った従来のテストで妥当性確認と最終的に市場に流出したメトリクスの比較を行った。

従来テストは 13 プロジェクトで実施したもの、シナリオテストはそれとは別の 8 プロジェクトで適用したものである。なお、それぞれのテストには新規開発プロジェクトを 1 つ含んでおり、残りはすべて派生開発プロジェクトで適用した。

表 74-2 シナリオテストと従来テストのメトリクス比較

メトリクス	従来テスト	シナリオテスト	考察
適用プロジェクト数 (プロジェクト)	13 プロジェクト (新規 1、派生 12)	8 プロジェクト (新規 1、派生 7)	
不具合密度 (抽出不具合数/KSloc)	0.20	0.69	従来テストよりも多くの 潜在不具合を抽出
不具合検出度 (抽出不具合数/テスト工 数)	15.59	15.88	テスト工数あたり同規模 の不具合抽出が可能
テスト進捗度 (実施テスト数/テスト工 数)	0.64	0.72	時間あたりのテスト消化 件数は差異なし
平均検出不具合数 (抽出不具合数/プロジェ クト)	2.46 件	5.88 件	検証ですり抜けた不具合 の流出防止向上
社外流出した不具合数 (件)	4 件	1 件	社外への不具合流出 75% 減

社外への不具合の流出抑止に関しては、従来テストにおける社外へ流出した不具合数は 4 件であったのに対し、シナリオテスト適用後は 1 件と 75%の削減を達成した。

妥当性確認プロセスにおける検出不具合数については、従来テストでは 1 プロジェクトあたり平均 2.46 件であったのに対し、シナリオテストでは平均 5.88 件と約 2.4 倍向上した。不具合密度は 0.20 から 0.69 と向上しており、従来テストよりも多くの潜在不具合を抽出できており、テスト品質を落とすことなく、妥当性確認プロセスにおける不具合流出に対する門番としての役割も向上した。

テストコストの指標として、テスト工数あたりの不具合検出数（不具合検出度）やテスト工数あたりの実施テスト数（テスト進捗度）がある。これらのメトリクスを比較したところ、従来テ

ストとシナリオテストではほぼ同じ数値となった。つまり、テスト工数やテスト進捗などのテストコストも従来テストとは乖離していない結果となった。

①前年度よりも流出不具合数よりも減らすことができた、②テスト品質は落としていない、③テストコストも従来テスト手法と大きく乖離していない、とすべての目標を達成することができた。

4.2. シナリオテストで検出できた不具合事例

次に、今回のシナリオテストにて検出できた特徴的な不具合例を表 74-3 に記す。

表 74-3 シナリオテストによる検出できた事例

項番	事例	検出できたポイント
①	LAN ポートに接続したケーブルの抜き差しで、映像の表示ができなくなる	ケーブル不良や接触不良のリンクチャタリング発生を想定して検出
②	特定のサーバ PC の再起動により、アプリケーションのサービスが起動しない	停電/復電想定とサーバスペック（低性能、標準性能、高性能）の組み合わせにて検出
③	装置初期状態の時に、特定の情報確認コマンドが動作しない	検証では状態が存在しない、装置の出荷状態のテストにて検出
④	特定の TFTP アプリケーションにて TFTP が動作しない	プロトコル互換性テストをユーザの利用頻度の高いアプリを選定し検出
⑤	CSV ファイル出力が表計算ソフトで正常な値が表示できない	日次/週次/月次トラフィックレポート報告を作る際のグラフ貼り付けを想定して検出

表 74-3 ①項は外部環境変化を意識したテストで発見できた。前工程の検証プロセスではポートのリンクアップ/ダウンの状態毎に、通信ができる/できない、リンク状態に応じたポート状態表示となることなど、1 つ 1 つの機能仕様を満足することのテスト観点が主体となっているが、シナリオテストでは外部環境が異常故障として変化した条件を組み合わせることで不具合の発見に至った。

表 74-3 ②項はサーバ用アプリケーションの日常業務を意識した再起動のテストにて発見できたものである。このシナリオテストのポイントは、サーバのスペックを顧客環境で考えられる複数のスペック（高性能、標準性能、低性能）を用意して行ったところ、起動時間のかかる低性能のサーバで不具合の検出に至った。

表 74-3 ③項は製造時の出荷検査工程を意識したテストで発見できた。出荷検査時で装置設定値が正しく設定されたことを確認する仕組みが実装されていなかった。製品出荷後はこの操作を行わないので顧客への影響はないが、「後工程はお客様」という言葉の通り、妥当性確認プロセスの後工程もすべてお客様であるという考えのもと、社内の製造後の出荷検査部門も顧客と考えてシナリオテストを実施したことにより、この仕様の抜けを検出できた。

表 74-3 ④項はアプリケーションの互換性のテストにて発見できた。検証ではプロトコル機能における設計との差異の確認が主体となっているため、プロトコルの解釈違いが発生しやすい。シナリオテストではアプリケーションとの互換性に着目していたため、特定のアプリケーションで動作しない不具合を発見することができた。

表 74-3 ⑤項もアプリケーションとの互換性のテストではあるが、直接的な互換性ではなく、製品がアウトプットする CSV ファイルをどのような業務、報告に用いるかに着目してシナリオテストを行ったところ、CSV ファイル内のカンマ (,) の扱いに関する不具合の検出に至った。

これら代表的な例にあるように、機能に着目した従来のテスト手法では見つけづらい発生条件の不具合を抽出できていることも、このシナリオテスト実施による効果と考えている。

4.3. テスト現場からの声

製品の不具合流出の抑制に効果があったという品質面の向上の他に、実際にこのテスト実施に携わったテスト担当者の意識向上を図ることができたという側面もあった。以下はテスト担当者の声を拾い上げた一例である。

① 異常系のテストケースがイメージしやすくなった

境界値分析や同値分割といったブラックボックステスト手法に則ったテストケースよりも、実運用で間違いやすい操作、たとえば入力を間違えて修正する手順や、一連の手順の順序を間違えて実行してしまったケースなどの方が、シナリオテストケース定義時に想定しやすいという効果があった。

② ユーザ目線で装置に触れることであるべき姿が見える

ユーザ目線で UI/UX を体感することを意識することで、繰り返し操作の面倒さや操作レスポンスの悪さなどのユーザビリティや、コマンド実行に順序性の制限があるケースに対する疑問などに気付けるようになり、あるべき装置仕様の正しい姿が見える効果があった。

③ 最終的なテスト結果での合否判定だけでなく、途中状態での状態確認も重要

顧客が使うシーンを想定したシナリオテストでは、いくつかのテストオペレーションを経て最終的な結果判定に導くテストケースが多い。このようなテストの実施において最終的な結果判定でのよし悪しだけでなく、テストオペレーション途中での状態確認も重要であることに気付く。たとえばシステム運用中に一連の操作を実行・完了するようなテストにおいては操作途中でもシステム運用に影響を与えてはいけないので、操作途中で異常なデータの出力がないか、想定外のログが記録されていないかなどのチェック意識も生まれる効果がある。

5. 今後の課題

5.1. シナリオテストケースの優先度と絞り込み

今後もシナリオテストケースを改善することで網羅性を高め、不具合の流出を防止し、品質向上の取り組みを続ける。一方でテスト工数は有限であり、また納期遵守も求められる。QCDをバランスよく保つためには、テストケースの優先度付けを行い、テストケースの絞り込みを行う必要が出てくる。優先度の算出は、ユーザの利用頻度の高い項目や、多くの開発が差分開発であることから変更部分の影響範囲から算出するのが自然な考えと思われる。しかし、ユーザの利用頻度をどのように算出するのか、また、ユーザシナリオベースにしたことで、機能的な項目がないため影響範囲の項目がわかりにくいなど優先度の決め方は簡単ではない。今後、優先度の決め方をどのようなパラメータと関連させて決定していくかが課題である。

5.2. 上流工程期間でのテスト設計の実施

妥当性確認のテストケースを作成するテスト設計の時期を、現在はV字モデルの右側、つまり下流工程で行っている。これをV字モデル左側の上流工程で実施することができれば、テスト設計段階で気づいた仕様の不備などを開発の早い段階で修正できる。品質を作り込むプロセスと並行して妥当性確認プロセスのテストケースを作成することで、仕様作成時の仕様チェックのフィードバックがより働き、よりコストをかけずに製品の品質向上が期待できる。(図 74-10)

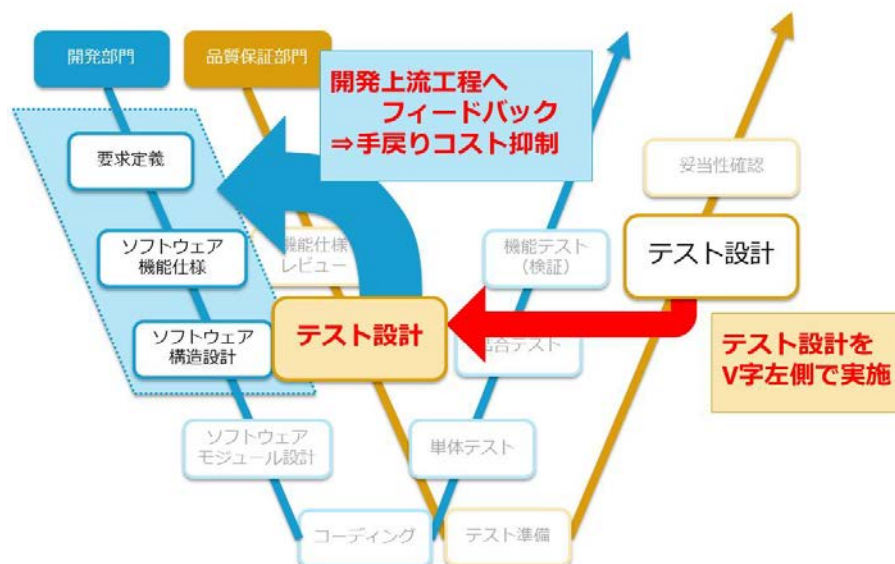


図 74-10 上流工程の品質改善

掲載されている会社名・製品名などは、各社の登録商標または商標です。
独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター (IPA/SEC)