

資料1：ODCの分類一覧＋補足説明

Defect Removal Activities : アクティビティ	どこのプロセスで発見したか? (発見工程)
1.Design Rev (デザインレビュー)	仕様のレビュー
2.Code Inspection (コードインスペクション)	ソースコードのレビュー
3.Unit test (ユニットテスト)	関数単体テスト
4.Function Test (機能テスト)	結合テスト
5.System Test (システムテスト)	システムテスト

Triggers : トリガー	不具合発生時の操作状況や環境、条件 (刺激)
1.Design Conformance (仕様適合)	仕様と整合が取れているか
2.Logic/Flow (論理/フロー)	フローそのものが、論理的におかしい
3.Backward Compatibility (旧バージョン互換性)	旧バージョンで動作していたものが、現在のバージョンで動作しなくなった
4.Lateral Compatibility (一般的互換性)	一般的な他のシステムとのインターフェースを利用する際の問題
5.Concurrency (並行性)	マルチタスクに関わる問題
6.Internal Document (内部文書)	記述漏れや記述誤りなど、ドキュメントの問題
7.Language Dependency (言語依存性)	文法や初期値、メモリ管理など、プログラミング言語に起因する問題
8.Side Effect (副作用)	例えば、バグ修正による新たなバグ
9.Rare Situations (レアケース)	めったに発生しない
10.Simple Path (通常使用)	単純な抜け漏れ、通常利用ができない
11.Complex Path (複合条件での使用)	前後条件を考慮できていない
12.Coverage (カバレッジ)	例えばエラー処理や例外処理など、想定されるケースがない
13.Variation (バリエーション)	例えば同値や境界などパラメータの組合せによる問題
14.Sequencing (連続操作)	繰り返し操作による問題
15.Interaction (相互作用)	各機能が独立して実行すると正常に実行されるが、特定の組み合わせで失敗するなど
16.Workload/Stress (負荷/ストレス)	負荷状態での問題
17.Recovery/Exception (リカバリ/例外)	例外処理のリカバリ対応の問題
18.Startup/Restart (スタートアップ/リスタート)	初期化や再起動時の問題
19.Hardware Configuration (ハードウェア構成)	特定のハードウェアに問題があった場合
20.Software Configuration (ソフトウェア構成)	特定のソフトウェアに問題があった場合
21.Blocked Test (previously Normal Mode) (ブロックされたテスト)	他の要因でテスト不可状態になった場合

Impact : 影響	どのような影響を及ぼすか?
1.Installability (インストール)	インストールが完了できない
2.Serviceability (保守性)	例えば、エラー番号だけでなく、どんな障害が発生しているかわからない
3.Standards (標準)	業界標準や社内基準などに準拠しない
4.Integrity/Security (完全性/セキュリティ)	不注意または悪意のある破壊、改ざん、または漏洩から、制御やデータを保護できない
5.Migration (マイグレーション)	アップグレードなどシステム移行できない
6.Reliability (信頼性)	安定して期待された役割を果たすことができない
7.Performance (パフォーマンス)	速度など性能が出ない
8.Documentation (ドキュメンテーション)	ユーザマニュアル・仕様書等に記載がないなど、説明がない
9.Requirements (必要条件)	製品が顧客の明示する期待を満たしていない
10.Maintenance (メンテナンス)	例えばスパゲッティコードなど、修正が容易にできない
11.Usability (ユーザビリティ)	使い勝手が悪い
12.Accessibility (アクセシビリティ)	例えば障害のある人に情報を提供できなくなっている
13.Capability (ケイパビリティ)	例えば、保存ボタンがあるのに保存できないなど、要求を処理する能力がない

Target : ターゲット	修正の対象。どこが悪かったのか?
1.Requirements (要件)	顧客の要件
2.Design (仕様・設計)	(同左)
3.Code (コーディング)	(同左)
4.Build/Package (ビルド)	(同左)
5.Information Development (メッセージ)	ユーザへの情報提供
6.National Language Support (言語)	マルチ言語対応

Defect Type : 欠陥の種類	何を間違ったのか?
1.Assign/Init (割当/初期値)	変数の値
2.Checking (チェック)	条件文のチェック漏れ
3.Alg/Method (アルゴリズム/実装方法)	(同左)
4.Func/Class/Object (関数/クラス/オブジェクト)	(同左)
5.Timing/Serial (タイミング/シリアル)	(同左)
6.Interface/O-O Messages (インターフェース/オブジェクト間メッセージ)	呼び出しミスなど
7.Relationship (関係性)	インスタンス化できない、クラス間の継承関係の問題など

Qualifier : 不具合の属性	どのように間違ったのか? なぜ欠陥が入り込んだのか、その原因
1.Missing (抜け漏れ)	抜け、漏れ
2.Incorrect (不正確/誤り)	不正確、誤り
3.Extraneous (無関係なものが混在)	例えば消し忘れなど、紛れ込んだ異物が原因

AGE : 新規性	どの時点で間違ったのか?
1.Base (従来)	変更などしていない既存のもの
2.New (新規)	新規に作成したもの
3.Rewritten (書き直した際)	再設計または古い機能の書きなおしを行ったもの
4.ReFixed (修正時)	不具合の修正を行った物

Source : ソース (出所)	発生源はどこか?
1.Developed In-House (自社開発)	自社開発
2.Reused From Library (ライブラリからの再利用)	ライブラリからの再利用
3.Outsourced (アウトソーシング)	アウトソーシング (協力会社など)
4.Ported (ポータリング)	移植元、母体

引用元: IBM Research (<http://www.research.ibm.com/softeng/ODC/DETODC.HTM>)

演習1の課題：不具合報告書の一例を、ODCで分類せよ。

プロジェクト名	「話題沸騰ポット」の開発		
表題	蓋を開けても、沸騰しない		
管理No.	システムテストの問題番号:31 (通番:931)		
不具合内容	検出日時	2011/7/21	
	検出者	テス太郎	検出工程 システムテスト
	不具合箇所・Ver.	蓋センサ機能	
	不具合詳細(現象)	蓋をして、電気コンセントを挿した後に電源を入れたが、沸騰が開始されない。(蓋を開けたら、沸騰開始された)	
	添付資料(有・無)	無	
	影響	蓋を開けたまま沸騰行為を行ってしまうため、大変危険である。これは、社内の安全基準を満たさない。	
原因・影響	原因	蓋センサのON/OFFのロジックを間違っており、逆になっていた。	
	修正箇所・Ver.	ソースコード	
	添付資料(有・無)	無	
	主な原因分類	自社開発のソフトに作り込まれた不具合である	
対応内容	ヘッダファイルの定義を修正する		



【分類結果】

Defect Removal Activities :アクティビティ	
どこのプロセスで発見したか?(発見工程)	
Triggers :トリガー	
不具合発生時の操作状況や環境、条件(刺激)	
Impact :影響	
どのような影響を及ぼすか?	
Target :ターゲット	
修正の対象。どこが悪かったのか?	
Defect Type :欠陥の種類	
何を間違ったのか?	
Qualifer :不具合の属性	
どのように間違ったのか?なぜ欠陥が入り込んだのか?	
AGE :新規性	
どの時点で間違ったのか?	
Source :ソース(出所)	
発生源はどこか?	

※資料1の「ODCの分類一覧+補足説明」を参考に、不具合情報を分類してください。

※各人で分類後、分類した結果について、グループ内でディスカッションしてください。

資料2：不具合リストと、ODC分類

問題番号	不具合内容	原因	対策	Activities	Trigger	Impact	Target	Defect Type	Qualifier	Age	Source	問題番号	欠陥場所	本来発見されるべき工程
1	満水センサの水位がONのとき、水位が上限を超えていると判断するが、一度上限と判定された後、OFFにならない。(水量が減っても、満水と判定される)	水位チェック処理時、満水センサのチェック実装漏れ。	満水センサがOFFになる判定を、水位チェック時に追加する。	5.System Test(システムテスト)	1.Design Conformance(仕様適合)	6. Reliability(信頼性)	3.Code(コーディング)	2. Checking(チェック)	1. Missing(抜け)	3. Rewritten(書き直した際)	4. Ported(ポーティング)	1	水位センサ処理	2.Code Inspection(コードインスペクション)
2	ボットの水温が、マイナス12度になると、温度表示をせずにエラー停止する。(ボタンを押しても反応しない状態)	センサの保証範囲を超過し、異常値が入力され、エラーとなった。	温度が異常値の場合には、「ERR」と表示するよう、仕様変更する。	5.System Test(システムテスト)	9.Rare Situations(レアケース)	2. Serviceability(保守性)	3.Code(コーディング)	1. Missing(抜け)	3. Rewritten(書き直した際)	3. Rewritten(書き直した際)	1. Developed In-House(自社開発)	2	エラー処理	2.Code Inspection(コードインスペクション)
3	蓋を閉めて3秒以内に給湯を開始すると、満水センサの水位がONのとき(異常値)でも、給湯できてしまう。	仕様書では、ポンプが動作できる条件に「水量が適切であること」とあるが、満水センサの検出は蓋を閉めて3秒経過後であったため。	給湯ボタンを押した時点で、水位を確認するように修正する。	5.System Test(システムテスト)	11.Complex Path(複合条件での使用)	9. Requirements(必要条件)	2.Design(仕様・設計)	3. Alg./Method(アルゴリズム/実装方法)	1. Missing(抜け)	2. New(新規)	1. Developed In-House(自社開発)	3	給湯処理	3.Unit test(ユニットテスト)
4	ボットの水温がマイナス温度のとき、温度表示がプラスになっている。(→5と表示)	出力時の形式(フォーマット)の指定誤り	フォーマットを修正する	5.System Test(システムテスト)	9.Rare Situations(レアケース)	11. Usability(ユーザビリティ)	3.Code(コーディング)	1. Assign/Init(割当/初期値)	2. Incorrect(不正確(誤り))	3. Rewritten(書き直した際)	2. Reused From Library(ライブラリからの再利用)	4	表示処理	
5	給湯を続けて水量が減り、空っぽになっても、給湯(ポンプ)が停止しない。	水量異常検出の、判定誤り。ゼロと1を、逆に判定していた。	判定処理を修正する。	5.System Test(システムテスト)	9.Rare Situations(レアケース)	9. Requirements(必要条件)	3.Code(コーディング)	2. Checking(チェック)	2. Incorrect(不正確(誤り))	2. New(新規)	1. Developed In-House(自社開発)	5	給湯処理	
6	ミルクモード(保温水量60度)で、給湯した温度を計測すると、65度になっている。	ヒーター操作量の制御周期を算出する計算式について、指定するパラメータの指定誤り。	指定するパラメータを修正する。	5.System Test(システムテスト)	1.Design Conformance(仕様適合)	6. Reliability(信頼性)	3.Code(コーディング)	3. Alg./Method(アルゴリズム/実装方法)	1. Missing(抜け)	1. Base(従来)	3. Outsourced(アウトソーシング)	6	ヒーター処理	
7	蓋を閉じると、保温中の温度に関わらず、毎回沸騰を行ってしまう。また、沸騰中は給湯できず、ミルクモードの利用時に不便。	仕様書通りに実装したが、利用時の要求にマッチしていなかった。	ミルクモードの場合は、蓋を閉じた際のお湯の温度が7度以下の場合に限り沸騰するよう、仕様を変更する。	5.System Test(システムテスト)	6.Internal Document(内部文書)	11. Usability(ユーザビリティ)	2.Design(仕様・設計)	3. Alg./Method(アルゴリズム/実装方法)	1. Missing(抜け)	4. ReFixed(修正時)	4. Ported(ポーティング)	7	沸騰処理	
8	牛乳を入れると…沸騰した後、使い物にならなくなる。	「ミルクモード」の説明不足。	取扱説明書の記載内容に、水以外を入れないように記載する。	5.System Test(システムテスト)	6.Internal Document(内部文書)	8. Documentation(ドキュメンテーション)	5.Information Development(開発)	4. Func./Class/Object(関数/クラス/オブジェクト)	2. Incorrect(不正確(誤り))	1. Base(従来)	1. Developed In-House(自社開発)	8	マニュアル	
9	ミルクモード(保温水量60度)で、給湯した温度を計測すると、97度になっている。	保温設定ボタンで、保温モード切替時に、目的の保温温度になるまで給湯禁止にしていなかった	保温モードに対応する温度になるまで給湯ボタンを押しても給湯されないようにする	5.System Test(システムテスト)	1.Design Conformance(仕様適合)	6. Reliability(信頼性)	1.Requirements(要件)	3. Alg./Method(アルゴリズム/実装方法)	2. Incorrect(不正確(誤り))	1. Base(従来)	3. Outsourced(アウトソーシング)	9	ヒーター処理	
10	3種類あるランプの点灯・消灯の論理が逆になっている。	1/0のポート(またはランプを制御するレジスタ)への書き込みの論理が1/0逆になっていた	ヘッダファイルの定義を修正する	5.System Test(システムテスト)	2.Logic/Flow(論理/フロー)	9. Requirements(必要条件)	3.Code(コーディング)	1. Assign/Init(割当/初期値)	2. Incorrect(不正確(誤り))	3. Rewritten(書き直した際)	2. Reused From Library(ライブラリからの再利用)	10	表示処理	
11	競合商品と比較して常温の水を沸騰させるまでの時間が2倍かかる	温度制御方式のロジック(式、パラメータ)に誤りがあった	温度制御方式のロジック(式、パラメータ)を修正する	5.System Test(システムテスト)	2.Logic/Flow(論理/フロー)	7. Performance(パフォーマンス)	3.Code(コーディング)	3. Alg./Method(アルゴリズム/実装方法)	2. Incorrect(不正確(誤り))	1. Base(従来)	3. Outsourced(アウトソーシング)	11	ヒーター処理	
12	給湯ボタンを押しても給湯されない場合がある。	組合せ条件で発生する。保温状態でもヒーターがonの場合は、沸騰行為と判断され給湯ボタンを押しても給湯出来なかった。	条件判定の修正	5.System Test(システムテスト)	11.Complex Path(複合条件での使用)	13. Capability(キャパビリティ)	3.Code(コーディング)	5. Timing/Seria(タイミング/シリアルライズ)	2. Incorrect(不正確(誤り))	2. New(新規)	1. Developed In-House(自社開発)	12	給湯処理	
13	給湯ボタンを押しても給湯されない場合がある。	センサ情報修正し入力していない。ボタンを押す力が弱い場合は、回路構成上給湯ボタンの信号がon/offを繰り返す。結果としてポンプがうまく動作せず、給湯出来なかった。	入力値を修正	5.System Test(システムテスト)	19.Hardware Configuration(ハードウェア構成)	13. Capability(キャパビリティ)	3.Code(コーディング)	6. Interface/O-O Messages(インターフェース/オブジェクト間メッセージ)	2. Incorrect(不正確(誤り))	2. New(新規)	1. Developed In-House(自社開発)	13	給湯処理	
14	沸騰行為が十分(沸騰前に保温に遷移したり、3分間の持続がない)に行われない。	PID制御のゲイン設定(増幅率設定)が不適切であったため何らかのエラーが発生していた。(このため、温度が上がらずエラー)	ゲイン設定の見直し	5.System Test(システムテスト)	2.Logic/Flow(論理/フロー)	6. Reliability(信頼性)	3.Code(コーディング)	3. Alg./Method(アルゴリズム/実装方法)	3. Extraneous(無関係なものが混在)	1. Base(従来)	3. Outsourced(アウトソーシング)	14	ヒーター処理	
15	蓋を開けても沸騰行為が中断されない	蓋が開いてもoff判定されなかった	判定処理の修正	5.System Test(システムテスト)	1.Design Conformance(仕様適合)	6. Reliability(信頼性)	3.Code(コーディング)	2. Checking(チェック)	1. Missing(抜け)	3. Rewritten(書き直した際)	4. Ported(ポーティング)	15	沸騰処理	
16	保温設定ボタンを押しても「ミルクモード」に変更しても水温が変わらない	保温モードと温度設定の不一致	判定処理の修正	5.System Test(システムテスト)	1.Design Conformance(仕様適合)	13. Capability(キャパビリティ)	3.Code(コーディング)	3. Alg./Method(アルゴリズム/実装方法)	2. Incorrect(不正確(誤り))	1. Base(従来)	3. Outsourced(アウトソーシング)	16	ヒーター処理	
17	水温表示が1の位はすべて0表示	pot-310-12の四捨五入の桁数指定が書かれていない	仕様を小数点第1位で四捨五入するように修正する	5.System Test(システムテスト)	6.Internal Document(内部文書)	8. Documentation(ドキュメンテーション)	2.Design(仕様・設計)	1. Assign/Init(割当/初期値)	1. Missing(抜け)	3. Rewritten(書き直した際)	2. Reused From Library(ライブラリからの再利用)	17	表示処理	
18	給湯中に沸騰ボタンを押すと、給湯後に沸騰行為に遷移する。	ソフト内部の沸騰開始要求フラグ(仮称)が保温状態で給湯中の沸騰ボタン押下でonとなり、給湯中は沸騰ボタンが無効という仕様を誤解釈していた。	沸騰開始要求フラグ(仮称)も、給湯中の沸騰ボタン操作ではoffのままとした	5.System Test(システムテスト)	10.Simple Path(通常使用)	11. Usability(ユーザビリティ)	3.Code(コーディング)	5. Timing/Seria(タイミング/シリアルライズ)	2. Incorrect(不正確(誤り))	3. Rewritten(書き直した際)	4. Ported(ポーティング)	18	沸騰処理	
19	保温行為中で給湯中でない場合に、沸騰ボタンを押したまま、沸騰ボタンを押し終わるまで沸騰行為に遷移しない。	沸騰ボタンの押下時間を押下開始時間と終了時間の差から求めている処理となっていた。	押下開始後100msec経過で押下されたと判断する処理とした	5.System Test(システムテスト)	20.Software Configuration(ソフトウェア構成)	11. Usability(ユーザビリティ)	3.Code(コーディング)	3. Alg./Method(アルゴリズム/実装方法)	2. Incorrect(不正確(誤り))	3. Rewritten(書き直した際)	4. Ported(ポーティング)	19	沸騰処理	
20	保温設定ボタンを押したまま、約200ms毎に高温→節約→ミルク→高温とモードが切り替わり続ける。	モード遷移要求フラグ(仮称)は保温設定ボタンの押下でonされ、モード遷移処理(押下判定100msとブザー鳴動100ms)完了でoffされる。しかし、offになった時点で保温設定ボタンが押下状態であるために、モード遷移要求フラグ(仮称)は再度onされてしまう。	保温設定ボタン一回の押下でモード遷移は一回とする。	5.System Test(システムテスト)	20.Software Configuration(ソフトウェア構成)	9. Requirements(必要条件)	3.Code(コーディング)	3. Alg./Method(アルゴリズム/実装方法)	2. Incorrect(不正確(誤り))	3. Rewritten(書き直した際)	1. Developed In-House(自社開発)	20	ボタン処理	

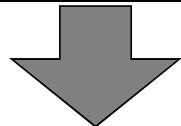
資料2：不具合リストと、ODC分類

問題番号	不具合内容	原因	対策	Activities	Trigger	Impact	Target	Defect Type	Qualifier	Age	Source	問題番号	欠陥場所	本来発見されるべき工程
21	実際の水位と水位センサが逆に表示される。水位センサ 第1水位→第4水位 第2水位→第3水位 第3水位→第2水位 第4水位→第4水位	I/Oポートへの接続が逆の順番になっていた	ヘッダファイルの定義を修正する	5.System Test(システムテスト)	19.Hardware Configuration(ハードウェア構成)	9. Requirements(必要条件)	3.Code(コーディング)	1. Assign/Init(割当/初期値) 2. Incorrect(不正確(誤り))	2. Incorrect(不正確(誤り))	3. Rewritten(書き直した際)	4. Ported(ポーティング)	21	水位センサ処理	
22	温度が実際の温度と異なる(一定温度の場合)	サーミスタ電圧から温度への変換式間違い	変換式を修正する	5.System Test(システムテスト)	1.Design Conformance(仕様適合)	6. Reliability(信頼性)	3.Code(コーディング)	3. Alg/Method(アルゴリズム/実装方)	2. Incorrect(不正確(誤り))	1. Base(従来)	3. Outsourced(アウトソーシング)	22	ヒーター処理	
23	温度が実際の温度と異なる(温度が変化する場合)。実際の水温の変化に対して計測した水温が分単位で遅れる。	温度の平均化フィルタの特定数大き過ぎ	変換式を修正する	5.System Test(システムテスト)	20.Software Configuration(ソフトウェア構成)	7. Performance(パフォーマンス)	3.Code(コーディング)	4. Func/Class/Object(関数/クラス/オブジェクト)	2. Incorrect(不正確(誤り))	1. Base(従来)	3. Outsourced(アウトソーシング)	23	ヒーター処理	
24	タイマをリセットした時のブザー音が、タイムアップした時のブザー音(100msec間隔で100msecを3回鳴らす)になっていた。	仕様誤認識により0min0secでのブザー音を同じにしていた	ブザー音を修正する	5.System Test(システムテスト)	8.Side Effect(副作用)	9. Requirements(必要条件)	3.Code(コーディング)	2. Checking(チェック)	3. Extraneous(無関係なものが混在)	4. ReFixed(修正時)	1. Developed In-House(自社開発)	24	ブザー処理	
25	取っ手がとつても熱くなる	蓋センサoff時の処理が入っていなかったため、蓋が開いたままでも沸騰行為を行っていた。	蓋センサoff時の処理を追加した。	5.System Test(システムテスト)	1.Design Conformance(仕様適合)	6. Reliability(信頼性)	3.Code(コーディング)	3. Alg/Method(アルゴリズム/実装方法)	1. Missing(抜け)	1. Base(従来)	3. Outsourced(アウトソーシング)	25	ヒーター処理	
26	容量の小さいシリーズは、急激に熱くなりすぎる	容量の小さいシリーズでも、従来の容量の大きいヒーターを採用したため、温度上昇が急激になった。	温度上昇率を規定した。	5.System Test(システムテスト)	3.Backward Compatibility(旧バージョン互換性)	6. Reliability(信頼性)	2.Design(仕様・設計)	1. Assign/Init(割当/初期値)	2. Incorrect(不正確(誤り))	1. Base(従来)	3. Outsourced(アウトソーシング)	26	ヒーター処理	
27	異臭がする	ドレッシングを入れて沸騰させた際に、110°Cでエラー検知せず、沸騰するまで温度上昇し、熱くなってゴム部分が溶けた	エラー検知機能の修正	5.System Test(システムテスト)	17.Recovery/Exception(リカバリー/例外)	6. Reliability(信頼性)	3.Code(コーディング)	1. Assign/Init(割当/初期値)	1. Missing(抜け)	1. Base(従来)	3. Outsourced(アウトソーシング)	27	ヒーター処理	
28	モ/によって沸騰するまでの時間にバラつきがある	ロットごとの部品のバラつきによる	部品をバラつきの小さいものにした	5.System Test(システムテスト)	19.Hardware Configuration(ハードウェア構成)	6. Reliability(信頼性)	1.Requirements(要件)	1. Assign/Init(割当/初期値)	2. Incorrect(不正確(誤り))	1. Base(従来)	3. Outsourced(アウトソーシング)	28	ヒーター処理	
29	給湯中に蓋を開けても給湯が止まらない。	仕様の給湯を停止させる条件に蓋を開けるという項目が抜けていた。	仕様を修正	5.System Test(システムテスト)	10.Simple Path(通常使用)	9. Requirements(必要条件)	1.Requirements(要件)	2. Checking(チェック)	1. Missing(抜け)	2. New(新規)	1. Developed In-House(自社開発)	29	給湯処理	
30	給湯ロック機能の初期状態がロック解除となっている	仕様書上、給湯ロック機能の初期状態がロック解除となっていた。	仕様を修正	5.System Test(システムテスト)	10.Simple Path(通常使用)	9. Requirements(必要条件)	1.Requirements(要件)	2. Checking(チェック)	1. Missing(抜け)	2. New(新規)	1. Developed In-House(自社開発)	30	給湯処理	
31	蓋をして、電気コンセントを挿した後に電源を入れたが、沸騰が開始されない。(蓋を開けたら、沸騰開始された)	蓋センサのON/OFFのロジックを間違っており、逆になっていた。	ヘッダファイルの定義を修正する	5.System Test(システムテスト)	10.Simple Path(通常使用)	6. Reliability(信頼性)	3.Code(コーディング)	3. Alg/Method(アルゴリズム/実装方)	2. Incorrect(不正確(誤り))	1. Base(従来)	1. Developed In-House(自社開発)	31	センサ処理	
32	蓋を閉じた後は、ロック解除となり給湯ボタン押下のみで給湯が可能となっている。	仕様書上、給湯ロック機能の蓋を閉じた状態がロック解除となっていた。	仕様を修正	5.System Test(システムテスト)	4.Lateral Compatibility(一般的互換性)	9. Requirements(必要条件)	1.Requirements(要件)	2. Checking(チェック)	1. Missing(抜け)	2. New(新規)	1. Developed In-House(自社開発)	32	給湯処理	

演習2-①：プロダクトの問題傾向を考察

影響度：Impact	分類される問題番号を、記載する。	件数	評価(ワーストに×)
記載例1	#2、#4、#5	3件	
記載例2	(なし)	0件	
記載例3	#1、#3、#6、#7、#8	5件	×(悪)
1. Installability (インストール)			
2. Serviceability (保守性)			
3. Standards (標準)			
4. Integrity/Security (完全性/セキュリティ)			
5. Migration (マイグレーション)			
6. Reliability (信頼性)			
7. Performance (パフォーマンス)			
8. Documentation (ドキュメンテーション)			
9. Requirements (必要条件)			
10. Maintenance (メンテナンス)			
11. Usability (ユーザビリティ)			
12. Accessibility (アクセシビリティ)			
13. Capability (ケイパビリティ)			

コード番号を読み上げる人と、問題番号を記載する人を分担すれば、スムーズかも。



評価が「×(最も悪い)」分類項目に対して、さらに他の分類と組み合わせて傾向を把握します。他の分類の内訳を集計する際、別の集計用紙を利用してください。

分類	左記の中で、最も件数の多い分類項目	件数	メモ
Triggers: どのような状況で発生するか？	例) 1.Design Conformance(仕様適合)		
Target: どこが悪かったのか？			
Defect Type: 何を間違ったのか？			
Qualifer: どのように間違ったのか？			
Age: どの時点で間違ったのか？			
Source: 誰の物が間違ったのか？			



プロダクトの問題傾向について、考察した結果を記載する。

考察
例) XXXがXXX件(XXX%)と、最も多い傾向にある。XXXXを重点とした見直しが必要。

※結果をグループ発表いただきます。どんな傾向を発見したのか、教えてくださいー！

演習2-②:集計用紙

演習2-①で、評価が×(ワースト)の分類項目を対象に、他の分類についても集計ください

Triggers:トリガー	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1.仕様適合			
2.論理/フロー			
3.旧バージョン互換性			
4.一般的互換性			
5.並行性			
6.内部文書			
7.言語依存性			
8.副作用			
9.レアケース			
10.通常使用			
11.複合条件での使用			
12.カバレッジ			
13.バリエーション			
14.連続操作			
15.相互作用			
16.負荷/ストレス			
17.リカバリー/例外			
18.スタートアップ/リスタート			
19.ハードウェア構成			
20.ソフトウェア構成			
21.ブロックされたテスト			

Defect Type :欠陥の種類	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1. 割当/初期値			
2. チェック			
3.アルゴリズム/実装方法			
4.関数/クラス/オブジェクト			
5.タイミング/シリアルイズ			
6. インターフェース/オブジェクト間メッセージ			
7. 関係性			

Target :ターゲット	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1.要件			
2.仕様・設計			
3.コーディング			
4.ビルド			
5.メッセージ			
6.言語			

Qualifier :不具合の属性	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1. 抜け漏れ			
2. 不正確(誤り)			
3. 無関係なものが混在			

Source :ソース(出所)	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1. 自社開発			
2. ライブラリからの再利用			
3. アウトソーシング			
4. ポーティング			

AGE :新規性	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1. Base (従来)			
2. New (新規)			
3. Rewritten (書き直した際)			
4. ReFixed (修正時)			

資料3：アクティビティと、トリガーのマッピングの一例

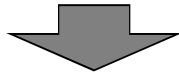
トリガー \ アクティビティ	1.Design Rev (デザインレビュー)	2.Code Inspection (コードインスペクション)	3.Unit test (ユニットテスト)	4.Function Test (機能テスト)	5.System Test (システムテスト)
1.Design Conformance(仕様適合)					
2.Logic/ Flow(論理/フロー)					
3.Backward Compatibility(旧バージョン互換性)					
4.Lateral Compatibility(一般的互換性)					
5.Concurrency(並行性)					
6.Internal Document(内部文書)					
7.Language Dependency(言語依存性)					
8.Side Effect(副作用)					
9.Rare Situations(レアケース)					
10.Simple Path(通常使用)					
11.Complex Path(複合条件での使用)					
12.Coverage(カバレッジ)					
13.Variation(バリエーション)					
14.Sequencing(連続操作)					
15.Interaction (相互作用)					
16.Workload/Stress(負荷/ストレス)					
17.Recovery/Exception(リカバリー/例外)					
18.Startup/Restart(スタートアップ/リスタート)					
19.Hardware Configuration(ハードウェア構成)					
20.Software Configuration(ソフトウェア構成)					
21.Blocked Test (previously Normal Mode)(ブロックされたテスト)					

※ヒント:ターゲット(修正対象)を参考にすると、対応するアクティビティが明確になるかも。

参考：IBM Research (<http://www.research.ibm.com/softeng/ODC/DETODC.HTM>)

演習 3-2 : 不具合を早い段階で低減させる施策を考察

見逃し工程	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1.Design Rev (デザインレビュー)			
2.Code Inspection(コード インスペクション)			
3.Unit test(ユニットテス ト)			
4.Function Test (機能テスト)			
5.System Test (システムテスト)			



評価が「×(最も悪い)」分類に対して、
さらに絞り込む分類を検討して、集計を実施。

	分類される問題番号を、記載する。	件数	評価(ワーストに×)
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			
12.			
13.			
14.			
15.			
16.			
17.			
18.			
19.			
20.			
21.			



現場の改善を目的として、具体的に実施すべき課題について
グループで検討した考察結果を、記載する。

考察	分類される問題番号を、記載する。
①主張：言いたいことの要点。	XXXXには、XXXの傾向があり、特に問題が集中している。
②根拠：事実に基づいたもの。(仮定とは、識別すること)	例) XXXとXXXで集計の結果、XXX件(XX%)と、最も比率が高い。
③提案：改善に向けて実施すべき課題	例)XXXXXについて、より詳細を確認し、重点的な見直し実施要。

※結果をグループ発表いただきます。なぜ、その分類を選んだのかも、発表してくださいー