

■ 欠陥混入メカニズム辞書

No	設計制約の変化	欠陥	ベースソフトの設計	設計の表現技法	ベースソフトの調査手順
1	Before) 機能が並列実行不可能 After) 機能が並列実行可能	共有データの排他処理漏れ	・機能間で共有しているデータ (外部変数)あり	-	1. 並行実行可能となった機能で、共有している外部変数を特定する 2. 1で特定した外部変数に対して、排他処理の追加要否を判断する
2	Before) オーバーランが発生する可能性のないバッファサイズとする After) バッファサイズ最小値の制約なし	バッファオーバーランのガード処理漏れ	・同じ種類のデータが格納されるバッファ(配列)が複数あり ・バッファ(配列)が保持されているユニットとは別ユニットから書き込みがされる箇所あり	・DFD	1. バッファオーバーランの可能性が発生したバッファを特定する 2. 1で特定したバッファを起点としてDFDを作成する 3. 2で作成したDFDから関連するバッファを特定する 4. 3で特定したバッファに対して、オーバーランのガード処理の追加要否を判断する
3	Before) 処理のキャンセル不可能 After) 処理のキャンセル可能	デッドロックの対策処理漏れ	・処理実行中の状態を複数のレイヤ(ユニット)でそれぞれ管理	・シーケンス ・状態遷移表	1. 処理中を示す状態を特定 2. 1で特定した状態について状態遷移表を作成する 3. 正常処理についてシーケンス図を作成する 4. 3で作成したシーケンス図からキャンセルが発生する可能性のあるタイミングを特定する 5. 2で作成した状態遷移表に対して、キャンセルトリガが増えることで、デッドロックが発生するケースがないか判断する
4	Before) 処理が異常終了することなし After) 処理が異常終了することあり	デッドロックの対策処理漏れ	・処理実行中の状態を複数のレイヤ(ユニット)でそれぞれ管理	・シーケンス ・状態遷移表	1. 処理中を示す状態を特定する 2. 1で特定した状態について状態遷移表を作成する 3. 正常処理についてシーケンス図を作成する 4. 3で作成したシーケンス図からキャンセルが発生する可能性のあるタイミングを特定する 5. 2で作成した状態遷移表に対して、キャンセルトリガが増えることで、デッドロックが発生するケースがないか判断する
5	Before) バッファのデータの歯抜けなし After) バッファのデータの歯抜けあり	データ取得時の無効値判定処理漏れ／有効値範囲判定処理誤り	・バッファのデータの並べ替え処理なし ・バッファ(配列)のデータ読み込み箇所が複数	-	1. データが歯抜けとなりうるバッファを特定する 2. 1で特定したバッファのデータ取得箇所を特定する 3. 2で特定した箇所に対して、無効値判定処理の追加要否を判断する
6	Before) バッファのデータの並び順に関する制約あり After) バッファのデータの並び順に関する制約なし	データ検索処理誤り	・バッファのデータの並べ替え処理なし ・バッファ(配列)のデータ読み込み箇所が複数	-	1. 並び順に関する制約がなくなるバッファを特定する 2. 1で特定したバッファのデータ取得箇所を特定する 3. 2で特定した箇所に対して、データ検索処理の変更要否を判断する
7	Before) データの有効区間が変化	データ取得時の無効値判定処理漏れ	・データの参照箇所が複数	タイミングチャート	1. 有効区間が変化したデータを特定する 2. 必要に応じて1.で特定したデータ更新タイミングを示すタイミングチャートを作成する 3. 1で特定したデータの参照箇所を特定する 4. 3で特定した箇所に対して、タイミングチャート等を活用し、無効値判定処理の追加要否を判断する