第 3 分科会　第 1 グループ（組込み見える化グループ）

# "組込み製品の品質を高めるための暗黙知の抽出・利用方法"

Method of extracting and using the tacit knowdge
for improving the quality in embedded software

主査　　飯泉　紀子　株式会社日立ハイテクノロジーズ
副主査　田所　孝文　株式会社山武
研究員　石原　鉄也　矢崎総業株式会社
　　清田　明宏　大崎電気工業株式会社
　　増田　耕弐　株式会社日立システムアンドサービス
　　杉山　篤志　富士ゼロックス株式会社

**研究概要**

ソフトウェア開発ではソフトウェアが作られていく過程で入り込んでしまう“バグ”の摘出はテスト工程に依存していることが多く、高品質のソフトウェアを開発するためには仕様検討段階のレビューでの摘出が望まれる。

一般的に、効果的なレビューを行うためには、レビュアー、技術者とも高度な専門知識を必要とする。しかし、優秀なレビュアーの論理的な思考、抽象化、概念化、洞察力は暗黙知であるため、レビューの効果が人に依存した状態からなかなか抜け出せない。そこで暗黙知を共有できることを目的に、その抽出と利用方法を提言する。

**Abstract**

The removal by the review of the specification examination stage is hoped for a lot of removals of "Bug" that enters it by the process from which software is made in the software development according to the test process to exist, and to develop the software of the high quality. In general, to review effective, reviewer, an engineer, and advanced expertise are needed. However, it is not possible to slip out the state for which the effect of the review depended on the person easily because thinking logical, abstracting, making to the concept, and the vision of excellent reviewer are tacit knowledge. Then, to share tacit knowledge, it proposes the extraction and use.

## １．背景

テスト工程でバグが検出されると手戻りが発生し、コスト超過、納期遅延のリスクが増大する。さらに、高機能化が進んでいることから、より多くのバグを上流工程で取り除いておくことがテスト工程まで見過ごされてしまうバグの削減に効果的である。

設計段階で指摘する機会としては、要求仕様, 製品(システム)仕様などの仕様書検討段階のレビ

ューがある．組込み製品としての特徴を鑑みた指摘は、この製品仕様から設計書に反映する段階である。

ただし、一般的に、レビューでの指摘については、以下のような問題がある。

- 文献や社内の規定として存在するチェック項目だけでは、指摘が不十分である。
- 特定のレビュアーと同じ指摘を他メンバーができない。または、指摘が設計者に正確に伝わらない。（経験をつんだレビュアーとそうでない人での指摘に差がある。）
- 指摘がほとんどされない。設計者による仕様説明会になってしまう。
- 設計者の回答に対して合否判断ができない。
- 過去の不具合事例が共有されていないために、過去にあったトラブルが見過ごされる。
- 設計者自体がレビュー慣れしていないため、レビューに臨むにあたって準備すること、品質向上のための設計の観点が理解されていない。

このような問題から、本研究では、特にレビュアーの観点を重視し、ソフトウェア仕様検討段階で利用するチェックリストを作成することで暗黙知を形式知にする活動を行った。

## ２．活動の目標

下流工程や出荷後に発生するトラブルや不良を上流工程における仕様検討（外部および内部仕様）段階であらかじめ洗い出すため、レビューの質を向上させたい。そのためには、現存するチェックリストの運用のみではなく、経験者がもっている暗黙知を有効利用できる環境作りが必要である。

そこで、『レビュアーが指摘しやすいよう暗黙知を文書化(チェックリスト化)し、共有することでトラブルを流出させないレビューができる』ようにするため、レビュー時に暗黙知を活用できるチェックリストを作成し、暗黙知が更新され形骸化しない運用方法の提案を目標とする。

## ３．活動内容

### ３－１．従来の問題点と今回の作成方法

従来あるチェックリストを適用・運用では何が問題であり、どこで弊害が発生するのか。問題点の提起、および対策方針からのチェックリスト作成方法を述べる。

#### ３－１－１．従来のチェックリストの問題点と方針

研究に際して参考としたチェックリスト（文献［1、2］）や社内で従来活用していたチェックリストでは、下記の問題点があった。

- 網羅性を高めるため、当たり前(形式知)となっているチェック項目が多い
- 応用性の高い(抽象的な)チェック項目が多く目的が不明確
- 組込み製品では技術革新が早く、チェックリストの陳腐化が早い

そのため本研究においては、以下に述べる３つのチェックリスト作成方針を定める。

①暗黙知を含んだチェック項目とし、網羅性を追及はしない

技術力の高いベテランレビュアーなどが個人的観点から注意すべき点や、過去のトラブル事例を基にチェック観点を追加（暗黙知を追加）することにより、後工程の手戻り削減に効果がある

ことは誰もが体験していることである。本研究では、これら観点のチェック項目のみで構成されるチェックリストを運用するだけでも効果はあると仮定し、網羅性を追及しないチェックリストを作成する。

②チェック項目だけでなく、その意図や観点を明確にする

従来あるチェックリストはチェック項目が羅列されただけのものであり、意図・観点を理解した上で使用する事が前提となったチェックリストが多数見られる。しかし本研究においては、暗黙知を形式知化することを目的にしていることから、暗黙知の意図・観点を十分理解してから使用する事が重要になってくる。そのため、レビュー時において「なぜその指摘をするのか」など、レビュアー側のチェックする意図・目的が明確になったチェック項目を作成する。

③定期的に形式知化・形骸化した項目を削除し、新規項目を追加する

作成するチェックリストはレビュー時において参照する運用を想定しているため、軽量であり全体が認識可能な項目数となっている必要がある。必要最低限の項目であり、ブラッシュアップされた項目とするため、定期的に形式知化・形骸化した項目を削除し、新たなチェック項目を追加する運用を提案する。その運用を前提とし、チェックリストの作成方針としては、現時点で実効性のあるチェック項目を厳選する。

### ３－１－２．チェックリストの作成方法

経験豊富な設計者は把握しているが他のメンバーと共有できていない知識（＝暗黙知）を共有することを目的としていることから、研究員全員でアイデアを出し合い、チェックリストのプロトタイプを作成し、議論を重ねた。(1)から(5)ならびに「図１：チェックリスト」の作成フローに示す方法で実施した。

(1) 項目案作成　：　最初は、１人あたり 30 件作成。その後は逐次追加する。
(2) 項目精査と分類　：　案の分類と、重複や内容確認を行う。
(3) リスト作成　：　検討した結果で、Verxx としてリストにする。
(4) ヒアリング　：　研究員が持ち帰り、３段階評価でそれぞれベテラン設計者 2-3 人にヒアリング。
(5) 集計、選定　：　低い点数となったもの削除を行うが、一人でも大変有効となっているものは点数だけでなく必要性を議論して、削除するか検討する。

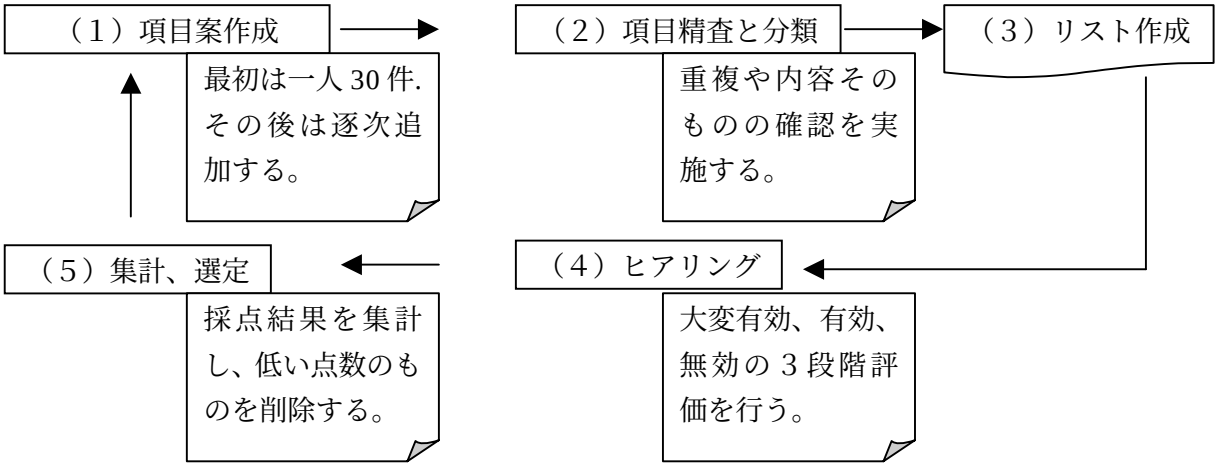


図１：チェックリストの作成フロー

### ３－２．チェックリストの紹介

文献[1、2]等で紹介されているチェックリストの各項目の要素として、カテゴリ、確認事項、判定基準が含まれており、今回作成したリストでもその要素は採用した。さらに、これまでのリストとの違いは以下の２つがあげられる。

１．前置き　：　レビューの流れにそって指摘できるようにするため、共有できている確認事項を前置きにし、その回答を受けた上で次の指摘にいけるようにする。

２．何をみたいのか　：　暗黙知の多くが、過去の不具合をベースとなっており、共有できてない場合が多いため、なぜこの確認をするのかを明記する。

リストは、従来のチェックリストにある３要素と上記２要素の５つの要素から構成される。

| | |
|---|---|
| カテゴリ | 要求仕様、過去トラブル、修正点、見積もりなど、項目の観点を分類したもの |
| 前置き | 共有できている指摘項目で、次の確認事項に対する指摘の回答をもらうための前提となる質問 |
| 確認事項 | 確認項目の実際の文章 |
| 何をみたいのか | その確認をする理由 |
| 判定基準 | 得られる回答として望ましい状態 |

上記の要素を含んだチェックリストの一部分を「表１：チェックリストの例」に示す。ここで例としているカテゴリは要求仕様である。要求仕様の分野で、トレーサビリティがとれているかどうかは、共有できている項目であり、これを前置きとしている。さらに、その確認ができた上で、次に何を確認したいのかの項目に対するように確認事項と判定基準を記載した。そこに暗黙知が含まれていることになる。また、確認事項の背景がわかれば、追加質問をしやすくなる。なお、注意点としては、判定基準は確認事項と対になっているが、記載されている基準を満たすことよりも、レビュアーが納得いく回答を引き出せればよいと考えているため、レビュアーがこの判断基準にとらわれないようにすることが必要である。

表１：チェックリストの例

| No | カテゴリ | 前置き | 確認事項 | 何をみたいのか | 判定基準 |
|---|---|---|---|---|---|
| M1 | 要求仕様 | ---- | トレーサビリティがとれているか？ | トレーサビリティがとれている。 | トレーサビリティがとれていることを証明するものがあること |
| 1 | 要求仕様 | M1 の回答があること | 複数の要求仕様書がある場合、整合が取れており、矛盾はないか？ | 記載内容の矛盾がないこと | ・顧客もしくは大本の要求仕様書にたどり着く流れは１つか？もし、２つある場合は、同じ記載になっているか。<br>・要求にない項目をつくっていないこ |

前置きの回答を受けた上で下の確認に入る。 → 確認事項の背景がわかれば、追加質問をしやすくなる。 → ○×だけでなく、レビュアーが納得いく回答を引き出す。

### ３－３．利用方法

チェックリストは研究員の所属組織の暗黙知を集結して作成したため、チェックリストを自社で導入する場合、独自に適用する暗黙知の追加および適用できない暗黙知の削除を行う必要がある。

### ３－３－１．使用方法　：　自社（QA）で適用するとすれば

以下の行為は全て QA が行い、レビューの指摘漏れを減らすことを目的としたチェックリストへの暗黙知の追加方法および運用方法を記載する。

【　レビュー前　】

ソフトウェアの設計変更を行う場合は、QA が始めに前回のレビュー議事録の指摘事項、テストにおけるバグおよびインシデント、市場に流出した不具合から失敗に対する原因のデータベースを作成し、データベースの分析および項目のスコアづけを行ってチェックリストへ暗黙知として追加し、チェックリストの更新をする。要求仕様と更新したチェックリストを照らし合わせて指摘事項を決める。スコアづけは、「３－１－２チェックリストの作成方法」に記載されている方法を用いる。

ソフトウェアの新規開発を行う場合は、前回のチェックリストを参考にチェック項目を決める。QA が、データベースの作成およびチェック項目の抽出を行うことで変更前のソフトウェアの問題点を把握でき、確認事項の要点を復習することもできるため、チェックリストにないチェック項目を要求仕様から見つけることが容易になる。

【　レビュー中　】

事前にチェック項目を開発へ知らせる場合は、QAが設計者の回答を納得できるようにチェックリストの内容および判定基準を加味した項目とする。レビューをする場合は、事前に用意したチェックリストの前置きを行い、QA自身が納得する回答を導き出す。

的を絞った指摘事項の回答を聞くことで、QAが明文化されてなかった設計方針を理解することができ、レビュー前には思いつかなかった指摘ができるようになる。

【　レビュー後　】

レビュー終了後もレビュー議事録の指摘事項のスコアづけを行い、チェックリストへ暗黙知として追加し、チェックリストの更新をする。暗黙知を追加したチェックリストをQAで回覧して情報を共有する。

今回のレビューで指摘事項の件数および暗黙知として追加された指摘事項の割合等の分析を行うことが出来れば、レビューの成否を議論できるきっかけとなる。

### ３－３－２．使用方法　：　自社（開発）で適用するとすれば

設計段階での使用方法としては、設計前にこれを利用して、注意点等の観点リストとして利用できる。元々、成果物の品質を確保するために作成されたチェックリストなので、設計者が設計前に知識として利用できれば、意識あわせをすることができる。

レビューを重ね、チェックリストを洗練していくことで、暗黙知が知識として明文化されていき、この暗黙知が記載されたチェックリストを利用することで、情報の共有に繋がるようになる。

### ３－４．チェックリストのライフサイクルについて

チェックリストが形骸化されずに継続して使用されるためには、チェック項目が新鮮であることがとても重要である。そのため、ここではチェックリストのライフサイクルを提案する。

具体的には半年や１年、プロジェクト完了時など、見直し時期を設け、チェック項目の実効性を再確認する。チェックリストからの新たな気付きや発見や、トラブル事例の再発防止策など、チェックリストへ追加すべきチェック項目が確認された場合、チェックリスト作成手順を再度実施し再構築を図る。

これにより、チェックリストの新規作成から始まり、プロジェクトでの活用、チェック項目の精査・追加、必要なチェックリストの抽出を経て、再度プロジェクトで活用できるという、チェックリストのライフサイクルが確立する。（図２参照）

このチェックリストのライフサイクルによって、形骸化しない新たな観点がチェック項目へ追加でき、時代や技術、チームノウハウと共に進化していくチェックリストの実現が可能となる。

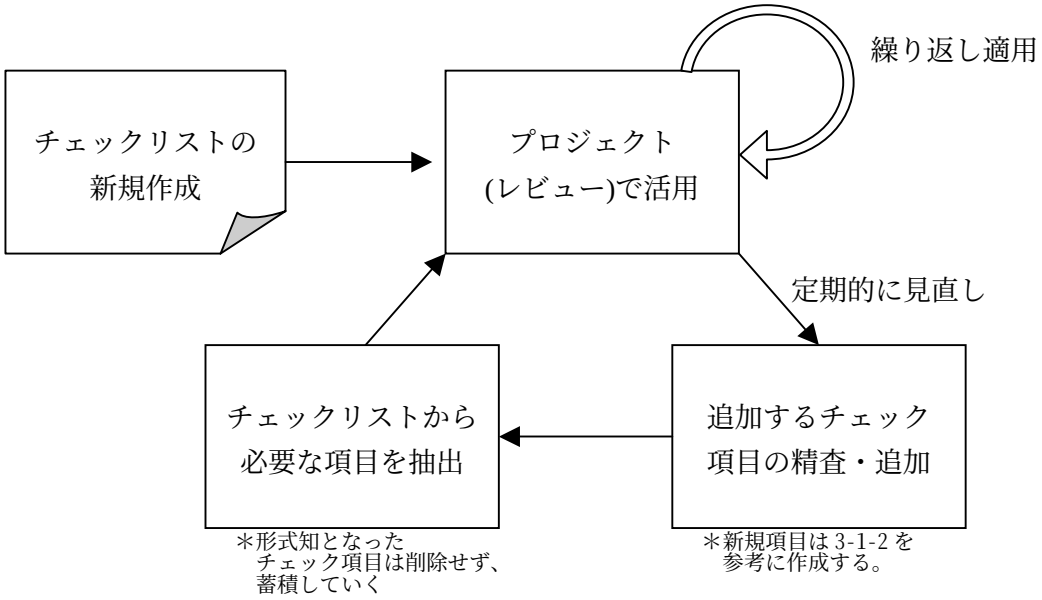


図 2 チェックリストのライフサイクル概念図

## ４．考察

本研究により、何人もの経験者の暗黙知を反映したリストを作成することができた。これを利用することで、上流工程で品質上重要な指摘ができるようなリストになると考えている。そして、ライフサイクルとして記載したように、改訂が必要であることを意識してほしい。

補足として、作成段階では以下の注意点があったので参考に記載する。

### ４－１．作成時の留意点，作成時の指摘から今後の作成時の参考点

暗黙知を形式知に変えることを意識したことや、様々なメーカーのメンバーが集まっていることから、作成段階で苦労した点がある。これらをまとめておき、今後のリスト改訂時に役立ててほしい。

(1)異なる製品と不具合事例の扱い

製品が異なれば、不具合事例は異なる。異なる不具合事例であっても、その不具合を再発させないための活動としてどのような視点を持つべきか検討することができる。このとき、一般的な言葉に直しながら事例を扱うと議論が進む。

(2)管理方法の違いの扱い

仕様書、フェーズ、問題点の管理方法が異なっているため、ある会社では当たり前に行われていることが他では行われていないことがある。今回の検討では、不具合管理表やフェーズ移行時の判定基準が該当した。当たり前のことでも、列挙しておき議論する必要がある。

(3)ソフトウェア作成の一般知識の扱い

フェールセーフ、フールプルーフなどの信頼性の考え方は広く知られているが、実際には適用できていないために不具合が発生している場合がある。製品毎に、フェールセーフが必要なのか、多少の性能劣化でも稼働しつづけることが必要なのかについては、「この考え方が入っ

ていること」という形で、当たり前であっても忘れずに実施することを徹底する。

(4)抽象的な項目から具体的な項目への分解

一般的な単語を用いると範囲が広がり、抽象的になりがちである。判定基準を意識して指摘項目を考えていくことや、抽象項目は前置きに回して、それに続く項目で有効な項目を探すと、具体的な項目になりやすい。

(5)同じ指摘が異なるカテゴリに含まれる場合の扱い

カテゴリ分類を終えると、各カテゴリの中に同じような項目が入り、かつ、判定基準が異なる場合がでてくる。使用時に項目漏れをなくすためにも、内容や判定基準が似ていても残しておいた方がよい。なお、一方に、他方の項目番号を書いておくことで、あえて同じ項目を残しているという意図が伝わる。

## ５．結論

当初の目的であった以下の項目を達成可能なチェックリストを作成した。

- 仕様検討段階での設計に対するチェックを前工程で実現すること
- 暗黙知を形式知にすること
- レビュアーの気付きを促すことができるチェックリストを作成すること

本研究活動で得たチェックリスト上の知見は、暗黙知を形にしたものであり、レビューに使用することはもちろん、暗黙知を持っていない設計者の知識獲得の参考資料としても使用できるものとなり、暗黙知を伝承するチェックリストを発案することが出来た。

本研究成果が、ソフトウェア開発の前工程での品質改善の向上に寄与し、ソフトウェア開発全体の品質向上に役立つことができれば幸いである。

## ６．今後の展開

今回、暗黙知を形式化するチェックリストの作成方法および管理方法の研究を行ったが、実証する段階までには至っていない。

チェックリストのライフサイクルを管理してレビュアーの質を向上させることは、高品質のソフトウェアを開発するだけでなくソフトウェア技術者の後継者育成に寄与できるのではないかと考える。

■参考文献

(1)情報処理推進機構(IPA) ソフトウェア・エンジニアリング・センター、IT プロジェクトの「見える化」下流工程編、日経 BP 社 、2006 年 6 月

(2)保田 勝通、ソフトウェア品質保証の考え方と実際、日科技連出版社、1995 年 11 月

**Appendix チェックリスト**

| 最終No | カテゴリNo | カテゴリ1 | カテゴリ2 | 前置き質問かどう | 前置き質問 | 指摘 | 内容(何をみたいのか) | 判定基準 |
|---|---|---|---|---|---|---|---|---|
| M1 | C1 | 要求仕様 | ーーー | 前置き質問 | ーーー | トレーサビリティがとれているか？ | ・トレーサビリティがとれている | トレーサビリティがとれていることを証明するものがあること |
| 1 | C1 | 要求仕様 | ーーー | ーーー | 前置きM1の回答があること | 複数の要求仕様書がある場合、整合が取れており、矛盾はないか？ | ・記載内容の矛盾がない | ・顧客もしくは大本の要求仕様書にたどり着く流れは1つであること<br>もし、2つある場合は、同じ記載になっていること<br>・要求にない項目を作っていないこと |
| 2 | C1 | 要求仕様 | ーーー | ーーー | 前置きM1の回答があること | 購入仕様とは別に客先と個別に調整定義した事項はあるか？ | ・仕様の抜け・漏れを防ぐ | ・記録が纏めてあること |
| M2 | C2 | 過去トラ | ーーー | 前置き質問 | ーーー | 過去トラブルのリストの中で、対象となる機能（仕様）はありますか？ | ・リストがある<br>・過去トラブルの対象となる機能（仕様）有無を明確にする | ・対象にする過去トラブルのリストがあること |
| 3 | C2 | 過去トラ | ーーー | ーーー | 前置きM2の回答があること | 過去トラブルは、どこまで対象にしていますか？ | ・対象範囲が一致している | ・対象範囲、重要度の認識があっていること<br>・対象とする過去トラブルをすべて確認していること |
| 4 | C2 | 過去トラ | ーーー | ーーー | 前置きM2の回答があること | 対象とする過去トラブルの対策はどうしましたか？ | ・再発防止ができている | ・過去の不具合と条件を同じにして確認をしているか．<br>実施できない場合はできない理由と，問題なさそうであると判断する理由を提示してあること．<br>・市場トラブルなどで、取り決めた再発防止策があるものは、その対応が実施されていること． |
| M3 | C3 | 修正点 | 機能 | 前置き質問 | ーーー | 修正機能は一覧として明確になっているか？ | ・機能修正点一覧（エンハンス内容一覧に含まれる場合はその中）で修正項目を明確にし、検討漏れを防ぐ<br>・修正量が少くても、変更点としてのリストアップがされている | ・機能修正点一覧が明確になっていること |
| M4 | C3 | 修正点 | 機能 | 前置き質問 | ーーー | 要求を実現する機能として、採用した理由が明確になっていますか？ | ・採用した理由がある<br>・必要のないところまで、変更していないかどうか確認する | ・採用した理由があること<br>・機能仕様書のレビュー議事録、関係チームとのレビュー議事録などがあること<br>・必要のないところを変更していないことを確認している資料があること |
| 5 | C3 | 修正点 | 機能 | ーーー | 前置きM3,M4の回答があること | 新機能導入のため、もしくは、保守性効率などのために、外部仕様を変えずに、内部動作を変えた機能は存在するか？ | ・内部動作を変えた機能が存在するかどうかを明確にする<br>・既存モジュールに影響があるのかかどうかを明確にする | ・存在する場合 → どこが、何のために、どのように変わったのかが明記された資料が存在すること。<br>・存在しない場合 → 設計書，モジュールのバージョンがかわっていないことを確認する（証明してもらう．） |
| M5 | C3 | 修正点 | ベース製品 | 前置き質問 | ーーー | 過去に類似する製品（モジュール）をベースとした開発品か？ | ・類似製品の有無を確認する | ・類似製品か、新規製品か、明確に答えられること |
| 6 | C3 | 修正点 | ベース製品 | ーーー | 前置きM5の回答があること | 利用する機能、変更追加（削除）する機能を明確にしているか？ | ・過去不具合事例<br>・母体製品の流用のために新製品で不具合が出る可能性があるので、母体の確認をする | ・母体製品の開発時に発生した不具合を調査し、本製品の品質向上観点に適用されていること |
| 7 | C3 | 修正点 | ベース製品 | ーーー | 前置きM5の回答があること | 母体製品の機能を利用する場合の使用上の条件、注意事項等はあるか？ | ・過去不具合事例<br>・母体製品の流用のために新製品で不具合が出る可能性があるので、母体の確認をする | ・流用母体や一部流用機能に、制限事項や前提条件、使用上の注意などが有る場合、本製品の制限事項とするのか、回避する予定などのリスクヘッジがなされていること。 |
| 8 | C3 | 修正点 | ベース製品 | ーーー | 前置きM5の回答があること | それぞれ、利用する機能、変更追加（削除）する機能で、タイミングなどで問題がないか？ | ・過去不具合事例<br>・母体製品の流用のために新製品で不具合が出る可能性があるので、母体の確認をする | ・データ、配列などの利用ロジックが、流用により不整合が発生しないか確認/検討できていること。<br>・処理タイミングをそのまま利用して、問題ないか確認できていること。<br>・共通Libを流用もしくは変更する場合、新規、既存両方の問題がないことを確認できていること |
| 9 | C3 | 修正点 | ベース製品 | ーーー | 前置きM5の回答があること | モジュールを追加および更新したことで、既存機能に影響がありますか？ | ・追加および更新したモジュールの影響を受ける機能、受けない機能を明確にする | ・モジュールの影響範囲が分かる関係図が存在すること。<br>・追加や更新したモジュールへ影響あるモジュールで構成する機能が明確になっていること。 |
| 10 | C3 | 修正点 | ベース製品 | ーーー | 前置きM5の回答があること | 修正対象となるモジュールと、他モジュールとのインターフェース確認は実施済みか？ | ・修正した内容に不整合が発生していないかチェックする（担当者の変更があった場合、各工程の初期と末期で、I/Fで接点があるモジュールに修正が発生した場合などは注意が必要） | ・同時期に修正されるモジュールやお互いに関わるインターフェースが明確になっていること。<br>・修正され、影響があるモジュール、インターフェイスの組み合わせ確認を行っていること。<br>・エビデンスが残っていること。 |
| M6 | C3 | 修正点 | HW | 前置き質問 | ーーー | ハード固有特有の特徴は考慮されているか？ | ・採用しているハードウェアの特徴を考えて設計しているかどうかを確認する | ・固有の特徴を述べられて、かつ対応を検討したことを示す資料があること |

| 最終No | カテゴリNo | カテゴリ1 | カテゴリ2 | 前置き質問かどうか | 前置き質問 | 指摘 | 内容(何をみたいのか) | 判定基準 |
|---|---|---|---|---|---|---|---|---|
| 11 | C3 | 修正点 | HW | ーーー | 前置きM6の回答があること | HWの変更の影響は明確なのか？ | ・ＯＳレベルの吸収、アプリレベルに影響有無を確認する | ・タイミング、データ量などが明確化されていること<br>・吸収方法、影響範囲が明確化されているこ<br>・過去の変更時のトラブルリストの検討結果があること |
| 12 | C3 | 修正点 | HW | ーーー | 前置きM6の回答があること | 以前からわかっているHWの特徴を考慮しているのか？ | ・以前からわかっているHWトラブル対応を確認する | ・以前から存在するHWの制限事項がそのまま引き継がれる場合、対策がなされていること。 |
| M7 | C4 | 見積もり、テスト | ーーー | 前置き質問 | ーーー | テスト計画は立案されているか？ | ・実施時期、規模、観点、問題点摘出目標を立てていることを確認する | ・テスト計画書があること、もしくはプロジェクトの開発計画書など、他の計画書の中にテスト計画相当の記述が含まれていること |
| 13 | C4 | 見積もり、テスト | ーーー | ーーー | 前置きM7の回答があること | 計画書において、変更コード量とバグ予測数、テスト件数は、過去プロジェクトの実績値と比較して、適正に設定されているか？ | ・(QCDの観点から)リスク対策がなされているか(必要か)事前にチェックする<br>・過去プロジェクトとの実績比較ができている | ・今回の見積もり基準（社内基準やFP法など）が明確になっていること<br>・特に過去実績(社内基準)と比較して、<br>⇒少ない場合は、理由は明確になっていること<br>⇒多い場合、従来どおりの開発体制で、品質を維持するために実施する対策があること |
| 14 | C4 | 見積もり、テスト | ーーー | ーーー | 前置きM7の回答があること | 評価,ユーザーの視点から設計の確認は行われているか？ | ・操作手順は、ユーザー視点でのチェックが行われている | ・ユーザーチェックテスト（第３者評価、設計者以外の評価者）を行う予定があること<br>・製造工程の関係者のテストがあること<br>・マニュアルチェックは行う予定があること<br>・エビデンスがあること |
| 15 | C4 | 見積もり、テスト | ーーー | ーーー | 前置きM7の回答があること | 実機動作によるテストを行いにくい部分のテストの把握と実施予定はあるのか？ | ・開発側の静的チェックによる判断が必要な部分を見極める<br>・必要な部分は、静的チェックを行って判断する<br>・テストできないプログラムをつくらせないようにする | ・レビューでテスト不可能を把握しているか？<br>→テストできないなら、レビューで問題点を取り除いているか？<br>・実機でテスト不可能な場合のテスト方法（ICE試験、C0/C1など）が明確になっていること<br>・ホワイトボックスのテスト結果が存在すること |
| 16 | C4 | 見積もり、テスト | ーーー | ーーー | 前置きM7の回答があること | テスト可能な仕様。設計となっているか？ | ・テストが出来ないプログラムは作らせない | ・ブラックボックスで確認できる部分の明確になっていること<br>・評価で確認してほしいと思っている部分を示す資料があること |
| M8 | C5 | Reset | 異常系 | 前置き質問 | ーーー | 正常系のチェックは全て完了しているか？ | ・正常系のチェック有無を確認する | 正常系のチェックがされている上で、<br>以下の異常系の確認がされていること |
| M9 | C5 | Reset | 異常系 | 前置き質問 | ーーー | リセット内容、リセットが発生する条件は明確になっているか？ | ・不完全なデータ受信や異常時の動作を確認する | ・何をリセットするのか明確になっていること<br>・正常値、異常値が明確になっているか(データ量・文字・桁数など)。<br>・入力データの正常範囲外のデータを受信した場合の処理が明確になっていること<br>・壊れたデータ（受信データ数情報と、実データ量が異なる場合など）受信時の処理も明確になっていること。 |
| 17 | C5 | Reset | 異常系 | ーーー | 前置きM8,M9の回答があること | 定義しているデータ、フラグが規定外に化けた場合に、どのような処理をするのか？ | ・不具合事例<br>・動作停止、オール初期化、復旧処理要求、など検討されている | ・パラメータの整合性チェックを行っていること<br>・チェックの結果、不整合が発覚した場合のシナリオ（動作停止 or 再起動～再復旧）が明確になっていること<br>・製品のポリシーとして、リセットの発生条件、停止条件が明確になっており、それに則っていること |
| 18 | C5 | Reset | 異常系 | ーーー | 前置きM8,M9の回答があること | 不完全なデータ受信や異常時の処理方式として、リセットが発生する条件は検討されているか？ | ・過去のトラブルから、リセットの発生条件などを見直している<br>出力側と整合がとれている | ・想定されるリセットが発生する条件、理由が明確になっている資料があること<br>・過去のトラブルから、発生条件を見直していること<br>・どのように検証したかを示す資料があること |
| 19 | C5 | Reset | 異常系 | ーーー | 前置きM8,M9の回答があること | 不完全なデータ受信や異常時の処理方式として、リセット以外の処理方法は検討されているのか？ | ・リセット以外の処理の検討がされている | ・リセット以外の処理があるかを確認できる資料があること<br>・過去のトラブルから、発生条件を見直していること |
| 20 | C5 | Reset | 異常系 | ーーー | 前置きM8,M9の回答があること | 想定しないリセット（WD、ノイズリセット）、処理途中の緊急停止（電源断含む）やキャンセルにより、リセット前後で動作（設定値）が変化することはあるか？ | ・動作（設定値）が変化した場合に問題が生じないかどうかを確認する<br>・問題が生じた場合に、他の装置（モジュール）が感知して復旧できるかどうかを確認する | ・状態遷移条件に抜けモレがないこと<br>（緊急停止、処理キャンセルのときの遷移は特に注意。）<br>・復帰処理が明確になっていること<br>・復帰できない場合に、他装置が感知するのか、自分でリセットをかけて、復旧するなどの処理が明確になっていること<br>・現時点で想定していない異常な状態になった場合（その他トラブル）の処理が明確になっていること<br><br>（HW/SWになんらかの異常がおきるから、想定していない状態になる。そういった場合、無応答やとんでもない値になったりするので、その場合の処理が明確になっていること） |

| 最終No | カテゴリNo | カテゴリ1 | カテゴリ2 | 前置き質問かどうか | 前置き質問 | 指摘 | 内容(何をみたいのか) | 判定基準 |
|---|---|---|---|---|---|---|---|---|
| M10 | C6 | 定型シーケンス | ーーー | 前置き質問 | ーーー | 初期化、WakeUpSleep、割り込みの確認はされているか | ・初期化、WakeUpSleep、割り込みの確認は終了している | ・初期化、WakeUpSleep、割り込みのシーケンスは確認できること<br>・WakeUp時に読み込むもしくは初期化するデータ、Sleep時に保存するデータは明確になっていること。<br>・状態遷移図は作成され、関係者によるレビューが実施されていること |
| 21 | C6 | 定型シーケンス | 初期化 | ーーー | 前置きM10の回答があること | 初期化項目と読み込み項目で、前任機との違いが明確になっているか？ | ・立ち上げシーケンスで、前任機との違いを把握する | ・前任機との変更点が明確になっていること<br>・電源投入時に初期化する項目と不揮発性メモリから読み込む項目の存在有無が明らかなこと<br>・変更箇所の局所確認と初期化シーケンスの通した状態の確認ができていること |
| 22 | C6 | 定型シーケンス | 初期化 | ーーー | 前置きM10の回答があること | データ化け、起動時の電源OFFなどの異常処理が発生した場合の処理は明確になっているか？ | ・データ化け、起動時の電源OFFなどの異常処理が発生した場合の処理を明確にする | ・データ化けがおきている場合の処理は確認できていること<br>・起動中の停止など、一度正常以外の立ち上げ方をしたあとでも、正常に立ち上げることができることを示す状態遷移図があること<br>・起動中の停止などで、データ書き込み異常などがおきている状態での起動処理が明確になっていること |
| 23 | C6 | 定型シーケンス | Wakeup/Sleep | ーーー | 前置きM10の回答があること | Sleep前に保存しなければいけないデータの扱いが明確になっているか？ | ・Sleep前に保存するデータと比較する（過不足が発生しているとトラブルになるため） | ・Sleep前に保存するデータと比較し、過不足が発生しないロジックを示すもの（仕様書）があること |
| 24 | C6 | 定型シーケンス | Wakeup/Sleep | ーーー | 前置きM10の回答があること | Sleepに入るまでの時間に誤りはないか．？ | ・時間に誤りがないこと（間違えるとトラブル発生になるため） | ・Sleep状態へ移行する条件、時間などの資料があること |
| 25 | C6 | 定型シーケンス | Wakeup/Sleep | ーーー | 前置きM10の回答があること | Wakup/Sleepともに割り込み処理は実施しているか？ | ・割り込みが正しく動作することを確認する | ・割りこみの評価計画があること<br>・外部要因（割り込み、イベント）が明記されている資料があること |
| 26 | C6 | 定型シーケンス | 割り込み処理 | ーーー | 前置きM10の回答があること | 割りこみの評価を実施しているか？ | ・割り込みに関する評価が実施されているかどうかを確認する | ・割り込み時、割り込み解除、割り込み禁止の処理の明確であること<br>・優先順位と戻り処理の明確であること<br>・多重割り込みの考慮されていること<br>・Wakup/Sleepともに割り込み処理の確認を実施していること |
| M11 | C7 | タイミング | ーーー | 前置き質問 | ーーー | モジュールを追加および更新した部分で、既存モジュールの処理時間をに影響があるのか。 | ・既存モジュールへの影響が無いことの確認する | ・処理時間に影響がある既存モジュールの存在有無を示す資料があること<br>・存在する場合、シーケンス図など、処理タイミングを検討した資料があること |
| 27 | C7 | タイミング | ーーー | ーーー | 前置きM11の回答があること | モジュールを追加および更新し、処理時間が変更したことで、カウントタイミングなどの基本処理に影響がないか？ | ・処理変更に伴う処理時間の増減を考慮しているかを確認する | ・モジュールを更新したことで、モジュールの処理時間がどのくらいになったか、予測もしくは実測値があること<br>・変更した処理時間のメインループへの影響がないことを確認できていること<br>・カウントタイミングに(増減の)マージンは取れていること |
| 28 | C7 | タイミング | ーーー | ーーー | 前置きM11の回答があること | モジュール間で同期ズレ（時刻等）が生じた場合に、補正する必要はあるか？ | ・補正が必要かどうか確認する<br>・必要な場合は、条件が定義されているかも確認する | ・同期ズレを補正する必要があるかどうか見極められていること<br>・補正した結果の影響は見極められていること<br>・時間などで前に戻るような処理で、問題が生じるかどうか見極められていること<br>・補正条件の定義があること |
| M12 | C8 | I/F | ーーー | 前置き質問 | ーーー | 追加および更新したモジュールの変数、引数、返却値は明確になっているか？ | ・追加および更新したモジュールの変数が他の変数に干渉しないこと | 返却値を引数として渡されたモジュールの引数は仕様と合致していることを示す資料があること |
| M13 | C8 | I/F | ーーー | 前置き質問 | ーーー | 追加および更新したモジュールのI/Fは明確になっているか？ | ・追加および更新したモジュールのIn/Outが明確になっていること | In/Outがあるモジュール全てに対し、マトリクスが作成されていること |
| 29 | C8 | I/F | ーーー | ーーー | 前置きM12,M13の回答があること | 追加された変数、引数の返却値、受け渡し条件は明確になっていますか？ | ・追加変数が他の変数へ干渉していないかを確認する（特に、返却値のインシデントに注意する） | ・返却値を引数として渡されたモジュールの引数が仕様と合致していることを示す資料があること。<br>（C0，C1でもかまわない） |
| 30 | C8 | I/F | ーーー | ーーー | 前置きM12,M13の回答があること | 変数，引数の境界値、異常値のテストで、どのような値を使うつもりか？ | ・異常系テストの実施有無を確認する<br>・使うデータの種類を確認する | 境界値、異常値、データ化けなどの値を使用したテスト成績書があること |
| 31 | C8 | I/F | ーーー | ーーー | 前置きM12,M13の回答があること | 変数がメモリサイズを越えていないか？ | ・変数がメモリサイズを越えることがないか確認する | ・プログラムのチェックツールを実施していること<br>・可能であれば、子プロセスや、さらに受け渡し先のプロセスで不正なデータとならないかチェックしていること |
| 32 | C8 | I/F | ーーー | ーーー | 前置きM12,M13の回答があること | 接続先、接続元のI/Fにおいて、データ形式、有効データ範囲などのデータプロファイルは一致しているか | ・I/Fの修正により、In側(もしくは子プロセス)の内部処理において不正処理(想定外のデータ)とならない | ・プログラムのチェックツールを実施していること<br>・可能であれば、子プロセスや、さらに受け渡し先のプロセスで不正なデータとならないかチェックしていること |
| M14 | C9 | データ | 平行実施 | 前置き質問 | ーーー | 処理の追加により、既存モジュールと平行実施することになったモジュールがあるか？ | ・既存モジュールへの影響を確認する | ・存在有無を確認した結果資料や仕様書などのチェック結果があること。 |
| 33 | C9 | データ | 平行実施 | ーーー | 前置きM14の回答があること | 既存の他プロセスとの競合、既存平行実行、多重実行が存在する場合、排他処理などは実施しているのか？ | ・データを共有するような構成のアクセス権限の定義の確認する<br>（アクセス権限：書き込み可/不可、読み込みのみなど、トラブルになりやすい部分であ | ・アクセス権限の基準が策定され、どの定義が適用されるか明確になっていること。<br>・アクセス権限のテストのエビデンスや実行結果が存在すること。 |

| 最終No | カテゴリNo | カテゴリ1 | カテゴリ2 | 前置き質問かどう | 前置き質問 | 指摘 | 内容(何をみたいのか) | 判定基準 |
|---|---|---|---|---|---|---|---|---|
| M15 | C9 | データ | 既存機能への影響 | 前置き質問 | ーーー | 追加および更新したモジュールの確認しているか？ | ・データシーケンス図、タイムチャートは存在するかどうかを確認する<br>・追加モジュールが存在するかどうかを確認する | ・データシーケンス図、タイムチャートは存在するか。<br>・追加モジュールリストが存在すること |
| 34 | C9 | データ | 既存機能への影響 | ーーー | 前置きM15の回答があること | 追加および更新したモジュールと競合する可能性のあるモジュールとのテストを行ったか？ | ・既存モジュールへの影響が無いことの確認する | ・追加モジュールの動作タイミングと競合する対象（モジュール、メモリ空間、データ）が明確になっていること<br>・同時書き換え、同時利用のチェックを行っていること<br>・同時書き換えがないことが望ましいので、もし発生した場合の動作が明確になっていること（たとえば、リセット） |
| M16 | C10 | 不揮発性メモリ | ーーー | 前置き質問 | | メモリマップはありますか？ | ・存在していることを確認する | ・メモリマップが存在すること |
| 35 | C10 | 不揮発性メモリ | ーーー | ーーー | 前置きM16の回答があること | 不揮発性メモリへの書き込み読み出しの動作とデータ内容、タイミングの確認は正しいか？ | ・EEPROMとCPU等との通信動作と通信内容およびその順序を確認する | ・過負荷時や他モジュールの処理遅延によるタイミングのずれは考慮されていること<br>・同時読み出し、同時書き込みの対応は考慮されていること<br>・データが読み出せない、書き込めない（壊れていた場合）の対応は考慮されていること |
| 36 | C10 | 不揮発性メモリ | ーーー | ーーー | 前置きM16の回答があること | 追加および更新したモジュールにて、不揮発性メモリへのデータ格納場所が正しいか？ | ・特異処理において、データ格納場所がズレないことを確認する | ・追加したモジュールの不揮発性メモリへのアクセスが存在するかどうかが明確になっていること<br>・データサイズとアドレスの確認はできていること<br>・新規に使用する領域が未使用領域を使用していること |