

**ソフトウェア開発 設計・実装チェックシート**  
**Software Designe Designe and Gording point check list**

No.	Category	Item	Check point	Answer and Evidence (Pre)	Comments
PS-01	購入ソフト品質 Purchase software quality	購入ソフトウェア(OSやプロトコルスタック等も含む)の選定の方法や選定の基準は明確か Is the selection method and selection criteria for purchased software (including OS and protocol stack) clear?	購入ソフトウェアの選定について以下のような手順や基準は明確に決まっているか 1) 購入ソフトウェアの選定を行う部署 2) 購入ソフトウェアの選定の手順 3) 購入ソフトウェアの採用の可否を決定する基準 選定の手順や採否基準を記述してあるルールを確認し、今回の製品に採用されている購入ソフトウェアについて具体的な記録を元に実施状況を確認する。  Are the following procedures and standards for selecting purchased software clearly determined? 1) Department that selects purchased software 2) Purchase software selection procedure 3) Criteria for deciding whether to adopt purchased software Check the rules describing the selection procedure and acceptance criteria, and confirm the implementation status based on specific records for the purchased software used in this product.		
PS-02	購入ソフト品質 Purchase software quality	購入ソフトウェアの品質を確認する手順は明確か Is the procedure for checking the quality of purchased software clear?	選定した購入ソフトウェアの品質を確認する手順や製品への採用の可否を判断する基準は、明確になっているか。品質確認の手順や採否の判定基準を記述してあるルールを確認し、今回の製品に採用されている購入ソフトウェアについて、具体的な記録を元に実施状況を確認する。  Are the procedures for confirming the quality of the selected purchased software and the criteria for determining whether or not the product can be adopted clearly established? Check the rules describing the quality confirmation procedure and acceptance criteria, and confirm the implementation status of the purchased software used in this product based on specific records.		
PS-03	購入ソフト品質 Purchase software quality	購入ソフトウェアは、販売元から十分な技術サポートを得られる体制があるか Does the purchased software have sufficient technical support from the distributor?	購入ソフトウェアの製品への組み込みにあたって、購入ソフトウェアの販売元からの技術支援が十分に得られる状況はできているか。今回の製品に採用されている購入ソフトについて、販売元による技術支援の仕組みを確認し、具体的な記録を元に技術支援の実施状況を確認する。  Is there a situation where technical support from the vendor of the purchased software can be sufficiently obtained when incorporating the purchased software into the product? For the purchased software used in this product, confirm the mechanism of technical support by the vendor, and confirm the implementation status of technical support based on specific records.		
PS-04	購入ソフト品質 Purchase software quality	購入ソフトウェアは販売元との保守契約が締結されているか Does the purchased software have a maintenance contract with the vendor?	購入ソフトウェアの保守について、当社向けの製品が必要とする保守期間のあいだ、ソフトウェアの販売元が保守サービスを提供し続けられるような保守契約が存在しているか。今回の製品に採用されている購入ソフトウェアについて、具体的な保守契約の内容を確認する。  Regarding the maintenance of purchased software, is there a maintenance contract that allows software vendors to continue to provide maintenance services during the maintenance period required by products for the Company? Check the details of the specific maintenance contract for the purchased software used in this product.		
PS-05	購入ソフト品質 Purchase software quality	購入ソフトウェアは市場で十分な稼働実績があるか Does the purchased software have sufficient operation results in the market?	購入するソフトウェアの当該バージョン品は、他社での利用実績も含めて、市場で十分な利用実績がある事が確認されているか。今回の製品に採用されている購入ソフトについて、具体的な記録を元にその品質実績の状況を確認する。  Is it confirmed that the version of the software to be purchased has a sufficient track record in the market, including the track record in other companies? Regarding the purchased software used in this product, the status of its quality performance will be confirmed based on specific records.		
PS-06	購入ソフト品質 Purchase software quality	購入ソフトウェアは、最近1年のバージョンアップの状況から判断して十分に安定しているか Is the purchased software sufficiently stable judging from the latest version upgrade in the last year?	購入するソフトウェアは、直近の1年間に何回のバージョンアップを実施しているのか、状況が把握できているか。特に不具合修正のバージョンアップが頻繁に実施されている場合には、まだソフトウェアが十分に安定していない懸念が残る。  Do you know how many times the software you purchase has been upgraded in the last year? here is a concern that the software is not sufficiently stable, especially when bug fixes are frequently updated. In that case, confirm whether any countermeasures are being considered.		
PS-07	購入ソフト品質 Purchase software quality	購入ソフトウェアは製品の開発終了以降も潜在バグや脆弱性の情報を確認する仕組みができていますか Does the purchased software have a mechanism for checking information on potential bugs and vulnerabilities even after the end of product development?	購入ソフトウェアは、当社での製品向けソフトウェアの開発が終了した後も潜在バグや脆弱性の発見が続く。これらの潜在問題の情報を集めるため、販売元からの最新バージョンのソフトウェアのリリース情報を入手して、当社の製品に含まれる潜在問題を確認できる仕組みはできていますか。  With regard to purchased software, potential bugs and vulnerabilities continue to be discovered after the development of software for products at our company ends. In order to collect information on these potential problems, is there a system in place to obtain the latest software release information from the distributor and confirm the potential problems included in our products? (Bugs and vulnerabilities corrected after the version of the purchased software adopted by our company correspond to potential bugs of the version adopted by our company)		
OS-01	OSS品質 Open Source Software Quality	採用するOSSの選定の方法や基準は明確か(OSSにはOSやプロトコルスタック等も含む) Is the OSS selection method and criteria to be adopted clear? (OSS includes OS, protocol stack, etc.)	OSSの選定について以下のような手順や基準は明確に決まっているか 1) OSSの選定を行う部署 2) OSSの選定の手順 3) OSSの採用の可否を決定する基準 選定の手順や採否基準を記述してあるルールを確認し、今回の製品に採用されているOSSについて具体的な記録を元に実施状況を確認する。  Are the following procedures and standards for OSS selection clearly determined? 1) Department that selects OSS 2) OSS selection procedure 3) Criteria for deciding whether to adopt OSS Check the rules describing the selection procedure and acceptance criteria, and confirm the implementation status based on specific records for the OSS used in this product.		

OS-02	OSS品質 Open Source Software Uuality	採用するOSSは市場での稼働実績は十分にあるかOSSの品質確認の手順は明確か Does the OSS to be used have sufficient operation results in the market? Is the OSS quality confirmation procedure clear?	選定したOSSの品質を確認する手順や製品への採用の可否を判断する基準は、明確になっているか。品質確認の手順や採否の判定基準を記述してあるルールを確認し、今回の製品に採用されているOSSについて具体的な記録を元に実施状況を確認する。 Is the procedure for confirming the quality of the selected OSS and the criteria for determining whether or not it can be adopted for the product clear? Check the rules describing the quality confirmation procedure and acceptance criteria, and confirm the implementation status based on specific records for the OSS used in this product.	
OS-03	OSS品質 Open Source Software Uuality	採用するOSSは市場での稼働実績は十分にあるか Does the OSS to be used have sufficient experience in the market?	採用するOSSのバージョンは、他社での利用実績も含めて市場で十分な利用実績がある事が確認されているか。今回の製品に採用されているOSSについて、具体的な記録を元に状況を確認する。 Has the OSS version to be adopted been confirmed to have sufficient usage results in the market, including usage results at other companies? Check the status of OSS used in this product based on specific records.	
OS-04	OSS品質 Open Source Software Uuality	採用するOSSのライセンス条件による採用の可否を判断する基準は明確か Are the criteria for determining whether or not the OSS can be adopted based on the license conditions of the OSS being adopted clear?	OSSには様々なライセンス条件があるが、どの条件のライセンスならば当社の製品に採用して良いかを明確に規定したルールはあるか。例えば、GPLV3のOSSは使って良いか、GPLV2のOSSは使って良いか等の具体的なルースを確認する。 There are various license conditions for OSS, but is there a rule that clearly stipulates which licenses can be used for our products? For example, confirm the specific looseness such as whether to use GPLV3 OSS or GPLV2 OSS.	
OS-05	OSS品質 Open Source Software Uuality	採用するOSSのライセンス条件による採用の可否を確認する手順は明確か Is the procedure for confirming whether or not the OSS can be adopted based on the license conditions of the OSS being adopted clear	採用するOSSが、採用条件(QS-04項)を満たしている事を確認する手順(プロセス)は明確になっているか。例えば、OSSのライセンス条件の確認とその結果に元々採用可否の判断は何時、誰が、どんな方法で実施するのか？ Is the procedure (process) for confirming that the adopted OSS meets the employment conditions (paragraph QS-04) clear? For example, when and who will carry out the OSS license conditions confirmation and the decision on adoption based on the results?	
OS-06	OSS品質 Open Source Software Uuality	採用するOSSの要求するソース開示やライセンス表示に対する対応方法は明確か Is it clear how to deal with the source disclosure and license display required by the OSS to be adopted?	採用したOSSによってはソース開示やライセンス条項の表示を要求する物もある。そのようなライセンス条項がある場合に、その要求を満たすための手順が具体的に規定されている事を確認する。 Depending on the OSS adopted, there are things that require source disclosure and license terms to be displayed. If there is such a license clause, make sure that the procedure for satisfying the request is specified.	
OS-07	OSS品質 Open Source Software Uuality	採用するOSSの安定性を確認する方法は明確か Is the method of confirming the stability of the OSS adopted clear?	OSSは、初期開発では機能追加や不具合修正で頻りにバージョンアップが行われる。採用しようとしているOSSの安定性を確認する方法は明確になっているか。例えば、以下のような情報を集めてOSSの安定性を確認している事を、確認する。 1) 直近の1年間に何回のマイナーバージョンアップ(不具合修正)を実施しているか 2) 最近3年間に何回のメジャーバージョンアップ(機能追加)を実施しているか 3) 重大な脆弱性が発見されて修正される事態が何度も起きていないか OSS is frequently upgraded in initial development by adding functions and correcting bugs. Is it clear how to check the stability of the OSS you are trying to adopt? For example, confirm that OSS stability is confirmed by collecting the following information. 1) How many minor version upgrades (defects) have been implemented in the last year 2) How many major version upgrades (function additions) have been implemented in the last 3 years 3) Hasn't it happened many times that serious vulnerabilities have been discovered and fixed?	
FS-01	FreeSoft 品質 Free software quality	OSS以外のFree Softwareの選定の方法や採用可否の基準は明確か。(Free Softwareとは、例えばチップベンダ提供のサンプルドライバやライブラリ等、無償だが無保証で提供されるソフトウェアを指す) Is the selection method for Free Software other than OSS and the criteria for adoption acceptable? (Free Software refers to software provided free of charge but without warranty, such as sample drivers and libraries provided by chip vendors)	Free Softwareの選定を行う部署や、選定の手順、採用の可否を決定する基準は、明確になっているか。選定の手順や採否基準を記述してあるルールを確認し、今回の製品に採用されているFree Softwareについて具体的な記録を元に実施状況を確認する。 Are the departments that select Free Software, the selection procedure, and the criteria for deciding whether to adopt them clear? Check the rules describing the selection procedure and acceptance criteria, and confirm the implementation status based on specific records for Free Software used in this product.	
FS-02	FreeSoft 品質 Free software quality	FreeSoftwareの品質確認の手順は明確か Is FreeSoftware quality verification clear?	選定したFree Softwareの品質を確認する手順や製品への採用の可否を判断する基準は、明確になっているか。品質確認の手順や採否の判定基準を記述してあるルールを確認し、今回の製品に採用されているFree Softwareについて、具体的な記録を元に実施状況を確認する。 Are the procedures for confirming the quality of the selected Free Software and the criteria for determining whether or not it can be adopted for the product clearly established? Check the rules describing the quality confirmation procedure and acceptance criteria, and confirm the implementation status of the Free Software used in this product based on specific records.	
FS-03	FreeSoft 品質 Free software quality	Free softwareは市場での稼働実績は十分にあるか Does Free software have enough experience in the market?	採用するFreesoftwareは、他社での利用実績も含めて市場で十分な利用実績がある事は、確認されているか。今回の製品に採用されているFreesoftwareについて、具体的な記録を元に状況を確認する。 Has FreeSoftware to be adopted been confirmed to have sufficient usage results in the market, including usage results at other companies? Check the status of Freesoftware used in this product based on specific records.	

FS-04	FreeSoft 品質 Free software quality	Free Software の安定性を確認する方法は明確か Is there a clear way to check the stability of Free Software?	採用しようとしている Free Software の安定性を確認する方法は明確になっているか。例えば、以下のような情報を集めて Free Software の安定性を確認している事を、確認する。 1) 直近の1年間に何回のマイナーバージョンアップ(不具合修正)を実施しているか 2) 最近3年間に何回のメジャーバージョンアップ(機能追加)を実施しているか 3) 重大な脆弱性が発見されて修正される事態が何度も起きていないか  Is it clear how to check the stability of the Free Software you are trying to adopt? For example, confirm that the stability of Free Software is confirmed by collecting the following information. 1) How many minor version upgrades (defects) have been implemented in the last year 2) How many major version upgrades (function additions) have been implemented in the last 3 years 3) Hasn't it happened many times that serious vulnerabilities have been discovered and fixed?		
IS-01	内作ソフト品質 (FROM) In-house software quality (FROM)	FROMアクセス処理には必要な排他制御が搭載されているか Does the FROM access process have the necessary exclusive control?	複数のタスクがアクセスするFROM領域に対しては、同時更新を禁止したり更新中の参照を禁止したりするなどの排他制御が設計/実装されている事を確認する。  Confirm that exclusive control such as prohibiting simultaneous update or prohibiting reference during update is designed / implemented for the FROM area accessed by multiple tasks.		
IS-02	内作ソフト品質 (FROM) In-house software quality (FROM)	FROMへの遅いアクセス時間が本来の処理に悪影響を起こさないか Is slow access time to FROM negatively affecting the original process?	FROMはアクセス時間が掛かる場合が多く、ソフトウェアの処理に影響を与える事が多い。FROMへのアクセス時間(特にWrite時)の最悪値(規定上の最大回数の書き込みを終えた後の値)がチップのハードウェア仕様から計算され、ソフトウェア処理時間の設計の時に問題無い事が検討されている事を確認する。  FROM often takes access time and often affects software processing. The worst value of the access time to FROM (especially when writing, the value after the specified maximum number of writes) is calculated from the hardware specifications of the chip, and there is no problem when designing the software processing time Confirm that is being considered.		
IS-03	内作ソフト品質 (FROM) In-house software quality (FROM)	FROMへのアクセスが起動処理に悪影響を与えないか Does access to FROM have a negative impact on startup processing?	起動処理では設定情報の読み出しやログ書き込み等のFROMアクセスが集中的に発生する事が多い。FROMの長いアクセス時間が、起動処理の実行タイミングに影響を与えない設計となっている事を確認する。  In the startup process, FROM access such as reading of setting information and log writing often occurs intensively. Check that the long access time of FROM is designed so as not to affect the execution timing of the startup process.		
IS-04	内作ソフト品質 (Timer) In-house software quality (Timer)	タイマとカウンタは重要な管理対象になっているか Are timers and counters important management targets?	プログラム内で利用されているタイマとカウンタは、一覧表にするなどの方法で設計・実装の状況を漏れなく確認できるような管理がされている事を確認する。  Check that the timers and counters used in the program are managed so that the status of design and implementation can be confirmed without fail by listing them.		
IS-05	内作ソフト品質 (Timer) In-house software quality (Timer)	タイマとカウンタのデータ型とサイズは適切な設計になっているか Is the data type and size of the timer and counter properly designed?	タイマーとカウンタを保持する変数は、取りうる最大値が定められていて、その値を格納できるデータ型とサイズとなっている事が確認されている事を確認する。  Check that the variables that hold the timer and counter have a maximum value that can be taken, and that they have a data type and size that can store that value.		
IS-06	内作ソフト品質 (Timer) In-house software quality (Timer)	タイマとカウンタはロールアップ処理が適切に設計・実装されているか Are timers and counters properly designed and implemented for rollup processing?	ロールアップする可能性のあるタイマとカウンタが一覧表等で洗い出されていて、ロールアップが発生した場合の処理が適切になっている事が確認されているか。  Are the timers and counters that may be rolled up identified in a list, etc., and confirmed that the processing when a rollup occurs is appropriate.		
IS-07	内作ソフト品質 (Timer) In-house software quality (Timer)	タイマとカウンタがロールアップする場合には、その処理が実機でテストされているか If the timer and counter roll up, is the process tested on the real machine?	ロールアップするタイマとカウンタが存在する場合、初期値をロールアップ直前の値に設定する方法で、ロールアップ処理が適切に動作する事がテストで確認されているか。  If there are timers and counters to roll up, have tests confirmed that the roll-up process works properly, such as by setting the initial value to the value immediately before roll-up?		
IS-08	内作ソフト品質 (mem leak) In-house software quality (mem leak)	動的に獲得や解放をする動的メモリは、重要な管理対象となっているか Is dynamic memory that is dynamically acquired and released an important management target?	プログラム内で利用される動的メモリは、OS機能が提供する動的メモリとアプリケーションに実装されている動的メモリがある。その両方について、一覧表にするなどして以下のような項目が確認できるように管理がされている事を確認する。 1) 初期状態 2) 最大サイズや枯渇監視の要否 3) 枯渇時の対処方法 4) 利用するタスク  Dynamic memory used in the program includes dynamic memory provided by OS functions and dynamic memory implemented in applications. Confirm that both are managed so that the following items can be confirmed by listing them. 1) Initial state 2) Maximum size and need for depletion monitoring 3) How to deal with exhaustion 4) Tasks to use		
IS-09	内作ソフト品質 (mem leak) In-house software quality (mem leak)	動的メモリについては、枯渇状況を監視する設計になっているか(必要な場合) Is dynamic memory designed to monitor depletion (if necessary)?	動的メモリを利用している場合、メモリの枯渇や予め定めたメモリ残量を下回る状況が起きる場合がある。このような状況を検出する監視処理の要否が検討されている事を確認する。  When dynamic memory is used, there may be a situation where memory is depleted or a predetermined remaining memory capacity is not reached. It is confirmed that the necessity of monitoring processing for detecting such a situation is being studied.		

IS-10	内作ソフト品質 (mem leak) In-house software quality (mem leak)	動的メモリについては、枯渇した時の回復処理が設計されているか(必要な場合) Is dynamic memory designed for recovery when it is depleted (if necessary)?	動的メモリが枯渇した場合や予め定めた残量以下になった場合に、その状況を回復する処理の必要は、検討されているか。(メモリ解放要求の発行や再起動による回復等) Is the necessity of processing to recover the situation when the dynamic memory is depleted or becomes less than the predetermined remaining amount considered? (Restoration by issuing a memory release request or restarting, etc.)		
IS-11	内作ソフト品質 (resource leak) In-house software quality (resource leak)	メモリ以外の動的資源についても重要管理対象となっているか(利用している場合) Whether dynamic resources other than memory are also subject to important management (if used)	メモリ以外に動的資源を使っている場合は、動的資源を一覧表にするなどして、最大量や枯渇監視の必要、枯渇した時の対処方法などが確認できるような管理がされている事を確認する。(メモリ以外の動的資源の例は、OS機能が提供するソケット、アプリケーションとして実装されている動的な登録・削除のある内部テーブル等) If dynamic resources other than memory are used, management is performed so that the maximum amount, the necessity of depletion monitoring, and the coping method when depleted can be checked by listing dynamic resources. Confirm that there is. (Examples of dynamic resources other than memory include sockets provided by OS functions, internal tables with dynamic registration / deletion implemented as applications, etc.)		
IS-12	内作ソフト品質 (resource leak) In-house software quality (resource leak)	メモリ以外の動的資源を使っているなら、枯渇状況を監視する設計になっているか(必要な場合) If dynamic resources other than memory are used, are they designed to monitor depletion (if necessary)?	メモリ以外の動的資源を利用している場合、枯渇した場合や、予め定めた残量以下になった場合の監視処理の必要は、検討されているか。 Whether dynamic monitoring resources other than memory are used, whether the necessity of monitoring processing is considered when exhausted or when the amount is below a predetermined remaining amount.		
IS-13	内作ソフト品質 (resource leak) In-house software quality (resource leak)	メモリ以外の動的資源を使っているなら、枯渇発生時の回復処理が設計されているか(必要な場合) If dynamic resources other than memory are used, is recovery processing designed when exhaustion occurs (if necessary)?	メモリ以外の動的資源を利用している場合、枯渇した場合や、予め定めた残量以下になった場合の監視処理の必要は、検討されているかどうかを確認する。(エラーログを記録するとか資源の回復処理を試みる等) If dynamic resources other than memory are used, whether or not monitoring processing is necessary when depleted or when the amount falls below a predetermined remaining amount is confirmed. (Record error log, try resource recovery process, etc.)		
IS-14	内作ソフト品質 (time out) In-house software quality (time out)	タイムアウト時間の設計方法は明確になっているか(必要な場合) Is the design method for the timeout period clear (if necessary)?	タイムアウト時間を決める方法が明確になっていて、誰がタイムアウト時間を設計しても同じ様にタイムアウト時間を設定できるようになっている事を確認する。例えば、考え得る最大の応答時間にXX%の余裕時間を加えて設定時間を決める、など具体的な方法が規定されているか。 Make sure that the method for determining the timeout period is clear and that anyone can design the timeout period in the same way. For example, is a specific method defined, such as determining the set time by adding a margin of XX% to the maximum possible response time?		
IS-15	内作ソフト品質 (resource leak) In-house software quality (resource leak)	機能追加や不具合修正の開発時に、タイムアウト値の設定に影響が無いか確認する手順はあるか Is there a procedure to check whether the timeout value setting is affected when adding functions or developing bug fixes?	タイムアウト時間は、機能追加や不具合修正等で各タスクの処理時間が変わる事による影響を受ける場合がある。変更開発のレビュー等で、タイムアウト時間を再確認する手順はあるか。特にFROMアクセスに関わる処理や、他機器との連携機能に変更があった場合に、タイムアウト時間の再確認を行う手順はあるか。 The timeout time may be affected by changes in the processing time of each task due to function addition or defect correction. Is there a procedure for reconfirming the timeout period in a change development review, etc.? Is there a procedure for reconfirming the timeout time especially when there is a change in the processing related to FROM access or the linkage function with other devices?		
IS-16	内作ソフト品質 (stat trans.) In-house software quality (stat trans.)	状態遷移の設計が必要な機能を洗い出す方法や基準は明確か Are the methods and criteria for identifying the functions that require state transition design clear?	状態遷移設計とそのレビューが必要な機能を洗い出す基準やルールは定められているか。例えば、状態とイベントの組み合わせの件数がXX件以上になるなら状態遷移設計のレビューを実施する、等状態遷移設計の必要性を判定する基準はあるか。 Are standards and rules established to identify functions that require state transition design and review? For example, if there are more than XX combinations of states and events, is there a criterion for judging the necessity of equi-state transition design, in which a state transition design review is performed?		
IS-17	内作ソフト品質 (stat trans.) In-house software quality (stat trans.)	状態遷移の設計の方法は明確か Is the state transition design method clear?	状態遷移の設計の方法は明確で文書化されているか。例えば、以下のような設計方法が具体的なルールやガイドラインとして作られているか。 1) 状態とイベントを洗い出し状態とイベントの対応を状態遷移表や状態遷移図で書き表す 2) 状態遷移図を使って必要な状態遷移を設計しドキュメントに書き表す 3) 状態遷移表を使って設計した状態遷移の網羅性をレビューする Is the state transition design method clear and documented? For example, are the following design methods created as specific rules and guidelines? 1) Identify the states and events and write the correspondence between the states and events in the state transition table and state transition diagram 2) Design the necessary state transitions using the state transition diagram and write it in the document 3) Review the completeness of the state transitions designed using the state transition table		
IS-18	内作ソフト品質 (stat trans.) In-house software quality (stat trans.)	状態遷移の設計をレビューする方法は明確になっているか Is it clear how to review the design of state transitions?	状態遷移の設計レビューでは、状態ハングが起きないような設計になっている事を確認するようなガイドラインがあるか。例えば以下のような確認項目が書かれたレビューの参考資料はあるか 状態やイベントの漏れ、状態を移動するイベントの漏れ、などの観点から確認する仕組みはあるか？ 1) 全ての状態とイベントが洗い出されているか 2) 全ての状態はその状態に入るイベントが存在するか 3) 全ての状態はその状態から抜け出るイベントが存在するか 4) 全ての状態はその状態から抜け出せなくなった時の回復処理があるか(タイムアウト等) Is there a guideline in the design review of state transitions to confirm that the design is such that state hangs do not occur? For example, is there any reference material for reviews with the following confirmation items? 1) Are all conditions and events identified? 2) Does every state have an event that enters that state? 3) Does every state have an event that exits that state? 4) Is there a recovery process when all the states can no longer get out of that state (timeout etc.)?		