仮日本語訳

ICT システムのための「データプロテクション・バイ・デザイン」のガイド

Guide to Data Protection by Design for ICT Systems

2019年5月31日

Personal Data Protection Commission (PDPC), Singapore

Privacy Commissioner for Personal Data (PCPD), Hong Kong

目次

1. はじめに

1.1 データプロテクション・バイ・デザインの概要

ICT（情報通信技術）システムのためのデータプロテクション・バイ・デザイン（DPbD）は、個人データを取り扱う ICT システムの開発において、開発プロセスの中にデータ保護対策を組み込むためのひとつのアプローチです。データ保護原則を最初から取り入れることで、個人データをより安全に保護し、優れたデータ管理を実践する文化を組織の中に醸成することができます。また、ICT システムの始まりから終わりまでのライフサイクルを通して DPbD を実践することにより、後でデータ保護機能を追加するのに比べ、不必要な遅れやコストを抑えることにも役立ちます。DPbD を後付けで対処するのではなく、組織の実働の中に組み込むことが大切です。

1.2 本ガイドの目的

このガイドは、ICT システムを設計および構築する際に、DPbD を適用したいと考える組織を支援することを目的としています。対象者はシステムもしくはソフトウェアの開発を実施する IT プロジェクトマネージャ、システム設計者、ソフトウェア開発者です。データ保護責任者（DPOs）は、どのように優れた実践がシステム開発プロセスの中に適用されるのかをよく理解できるようになります。このガイドでは次の事項が記されています。

・DPbD の原則
・ソフトウェア開発ライフサイクル（SDLC；Software Development Lifecycle）の各フェーズにおける DPbD の活動
・ICT システムの優れたデータ保護の取り組み

なお、このガイドで述べたヒントや優れた取り組みは網羅的なものではなく、すべての状況に関連しているわけではありません。それぞれの組織で、対象としているビジネスや運用状況に合わせて合理的かつ適切に実践することが必要です。

1．3　データプロテクション・バイ・デザインの原則

カナダのオンタリオ州の前情報プライバシー保護委員であるアン・カブキアン博士（Dr. Ann Cavoukian）が開発したプライバシー・バイ・デザインの7つの基本原則[1]はよく認知されており、多くのデータ保護の専門家がよく参照しています。

この7つの基本原則を参考にすると、ICT システム開発を進めるための適切かつ重要な7つの DPbD 原則は以下のようになります。

① 事前的かつ予防的

データ漏えいが起こる前にデータ保護リスクの評価、特定、管理、そして予防を行います。優れたデザインやデータ管理を実践することでリスクを最小限にできます。

② デフォルトとしてのデータ保護

データ保護対策はシステムのプロセスや機能に統合されていなければなりません。個人が自分の個人データを保護するのではなく、個人データを保護する手段はデフォルト設定として自動的に提供されていることが必要です。

③ エンド・ツー・エンドのセキュリティ

セキュリティ対策はソフトウェア開発ライフサイクル（SDLC）全体にわたって配慮する必要があります。個人データが収集されてからシステムで削除されるまで、SDLC の各フェーズに優れたセキュリティの機能や実践を組み込むことができます。

組織とベンダーがどのように連携するのか、またソフトウェア、ハードウエア、製品、サービス、プラットフォームなどの ICT システムの要素がどのように連携するのかという観点から、ユーザは「エンド・ツー・エンド」で徹底して検討を行う必要もあります。

この「エンド・ツー・エンド」の観点から脆弱な部分を探しだし、システムのセキュリティを強化する方法を評価してください。

④ データ最小化

個人データを扱うとき、「まずはデータを集め、あとで何をするかを考える」というアプローチをしないでください。データ最小化とは、目的達成に必要なだけの個人データを収集し、保存し、使用することを意味します。

注 1: https://www.ipc.on.ca/wp-content/uploads/resources/7foundational principles.pdf

⑤ ユーザ中心

ユーザ個人を中心にして考える、つまり個人の個人データの保護を目的としてICT システムを開発し実装してください。デフォルトを設定すると同時に、個人がその設定をカスタマイズできる機能や方法を提供してください。そのインタフェースはユーザフレンドリーでなくてはいけません。例えば、必要なときに通知される「ジャストインタイム」や階層化された通知などの機能を適用することです。

⑥ 透明性

何の個人データが収集され、どのように使われているのかをユーザ個人に積極的に知らせるようにしてください。さらに、第3者がその個人データを処理していることを知らせてください。そのような情報をユーザに提供するために適切な手段を選択して使用してください。これらはユーザとのインタラクションあるいはジャストインタイムの通知とは異なる観点です。

⑦ リスク最小化

DPbD の重要なポイントはデータ保護のリスクをシステマティックに特定して軽減することです。個人データが処理される際のシステムの適切な処置と、関連する ICT セキュリティ対策が設計され実装されていることで、リスクを減らすことができるのです。

1．4　データプロテクション・バイ・デザインと SDLC

ソフトウェア開発ライフサイクル（SDLC）の各フェーズにおける主要な DPbD の活動を以下に示します。

SDLC フェーズ	DPbD の活動
要件	ここから DPbD が始まります。データ保護の観点から不明確な目的あるいは合理的でない要件を評価して選別してください。そのような要件を削除します。
設計	DPbD 原則がシステムアーキテクチャレベルの設計に組み込まれていることを保証するための重要なフェーズです。
開発	システムが開発されるフェーズで、DPbD 原則が考慮され、その結果としてシステムのさまざまな機能や要素に反映されます。
試験	試験の結果から、前フェーズで定義された要件や DPbD の考慮事項が、システムに組み込まれていることを確認するチェックフェーズです。
移行	システムをリリースする準備フェーズです。リリース直前にシステムの処理や構成の最終チェックを行います。
保守	システムのセキュリティを維持し、移行後のシステムをより堅固にするためのフェーズです。組織がシステムを見直したり更新したりするときに、引き続き DPbD が適用されていることが必要です。

１．５ データプロテクション・バイ・デザインとアジャイルソフトウェア開発

前節に述べたソフトウェア開発モデルは一般に「ウォーターフォール」モデルと呼ばれています。ウォーターフォールモデルでは、システム開発のすべての要件が事前に整理され、SDLC のひとつのフェーズとして順番にかつほぼリニアにまとめて実行されます。

これに対し、アジャイルソフトウェア開発はソフトウェア開発のもうひとつの方法です。ウォーターフォールに比較すると、アジャイルソフトウェア開発はたくさんの開発サイクルがあるなかで、各サイクルに必要な要件を少しずつ満たしながら、繰り返し開発を進める方法です。

アジャイルの各サイクルは、ウォーターフォールの SDLC において部分的に関連するところがあり、このガイドで述べるグッドプラクティスはアジャイル開発モデルにも適用することができます。

実際、アジャイル開発では、基本的な機能設計や当初からの変更に対して、より柔軟により低コストで実現できる可能性があります。そのため、可能なら最初のサイクルで、すぐに構築できなくとも、DPbD 機能が考慮され計画されることで費用対効果が高くなると考えられます。

データ保護影響評価（DPIA；Data Protection Impact Assessments）について、アジャイルプロジェクトの初期段階でできるだけ多くの情報を利用して包括的な DPIA を実施することが望まれます。その後に続くサイクルのなかであるいは DPIA を繰り返し実施することで、DPIA による評価を更新することができます。

1．6　既存 ICT システムのためのデータプロテクション・バイ・デザイン

新たな ICT システムに DPbD を組み込むことが望ましいのですが、現実には少なくともいくつかの既存 ICT システムを組織が保有していることが多いと思います。それでは、既存 ICT システムのデータ保護を改善するために、組織は何ができるのでしょうか。

マリリン・プロシュ（Marilyn Prosch）准教授とアン・カブキアン（Ann Cavoukian）博士は既存 ICT システムのための「プライバシー・バイ・リデザイン」[2] の概念を開発しました。コンセプトは3つの R です。

・再考（Rethink）は、どんな個人データが収集され、その収集がほんとに必要か否かを検討してリスクを評価するなど、システムを徹底的に見直すことです。この再考段階で DPIA を実施するのが適切です。

・再設計（Redesign）は、個人データをより適切に保護し、以前に見つかったリスクを減らすことで、関連する DPbD のグッドプラクティスを実装することです。

・再生（Revive）は、データ保護を強化したシステムから新たに始めることです。

組織は、リエンジニアリングのプロセスは一度に実装するか、複数フェーズに渡って段階的に実装するかを決定します。

注 2：http://www.ontla.on.ca/library/repository/mon/25005/310082.pdf

2. ICT システムのためのデータプロテクション・バイ・デザインのグッドプラクティス

2.1 グッドプラクティス

この章では ICT システムのための DPbD のグッドプラクティス（優れた実践や取り組み）を紹介します。このグッドプラクティスはトピックごとにまとめられています。ユーザは関連するグッドプラクティスを選び、自分たちの ICT プロジェクトに適合させることが必要です。なお、ここにあげたグッドプラクティスはすべての領域をカバーしているわけではないので気をつけてください。

グッドプラクティスを適用する箇所を知ってもらうことでソフトウェア開発者をサポートするため、各グッドプラクティスと最も関連する SDLC のフェーズ、および一般的に使われる「3 階層（3-Tier）」のアーキテクチャ（プレゼンテーション、アプリケーション、データ）[4] において関連する層を表中のチェック印で明示しています。なお、グッドプラクティスと SDLC や層との関連付けは規定されたものではありません。

注 3：3階層（3-Tier）のプレゼンテーション（表現）層は情報の表示やユーザとの対話を担当し、アプリケーション（応用）層はアプリケーションのビジネスロジックを管理し、データ（DB）層はデータを保存し管理する場所である。

訳注：本注は原文では注 4 であるが、注 3 がないために項番を繰り上げた。以降も同様。

2．2　データ保護影響評価（DPIA）

DPIA は SDLC のどの時点でも実行されるかもしれないが、DPbD の本来の目的から言えば、新たな ICT システムの予備設計の時点がタイミング的によい。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1．開発の前に DPIA を実施 新 ICT システムの目的達成に必要な個人データのタイプと処理について厳密に評価します。 これは個人データに関する新 ICT システム設計のギャップやリスクを特定して評価するのに役立ちます。ICT システムの予備設計の変更や必要に応じて特定したギャップやリスクの軽減策を組み込むことも可能です。 詳しくは PDPC のデータ保護影響評価のためのガイド（Guide to Data Protection Impact Assessments）を参照のこと。	○	○				○			

2．3　ICT システムによる個人データの収集

収集しても使わない個人データを所有することで、組織は余計なリスクを背負うことになります。不必要な個人データの保護に必要なリソースは、最初から収集しないことで単純にカットすることができます。すなわち、もっとも良い収集の方法は、組織にとって真に必要な個人データは何かをまず考えることなのです。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1. 最小限の個人データを収集 収集する明確な目的があり使用する予定のある場合をのぞき、個人データを収集しないでください。各データ項目に対し、収集時に以下のことを確認してください。 ・組織の機能や活動に直接的に関連する法律的、合理的な目的があること。 ・目的を逸脱しないこと。 同じ目的のために異なるタイプの個人データを使う際には、最も機密性の低い個人データを収集してください（例えば、ユーザの正確な位置ではなく、おおよその位置データを収集）。	○	○				○	○	○	
2. どうしても必要なときには、個人の識別子情報（例えば個人識別番号）だけを収集 個人を直接識別する固有の値になるので、ほんとうに必要か否かについて、よく検討する必要があります。 シンガポールでいえば、法律で定められているとき、あるいは厳密に個人の身元を立証あるいは検証が必要なときは、国民登録番号カード（NRIC; National Registration Identification Card)だけを収集してください。 詳細な情報は、PDPC の NRIC や他の国民識別番号のための個人データ保護法に関するアドバイザリーガイドライン（Advisory Guidelines on the Personal Data Protection Act for NRIC and Other National Identification Number）を参照のこと。	○	○				○	○	○	

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
3. メタデータに注意 メタデータ(例えば、画像ファイルの EXIF データ)として、意識せずに個人データを収集してしまうことがあります。このようなデータは収集しないか、必要がないのであれば削除することを考えてください。	○	○				○		○	○
4. 個人データは継続的に収集せず、必要なときに収集 例えば、モバイルアプリでは、ユーザの位置情報を常に収集するのではなく、ほんとうに必要なときにだけユーザの位置を収集するオプションを提供してください。	○	○	○			○	○	○	
5. 個人データを自動的に収集せず、ユーザの入力で収集 便利さを好むユーザもいれば、介入されるのを嫌がるユーザもいる。例えば、モバイルアプリでは、自動的に位置の追尾を行うのではなく、ユーザに自分の位置情報を示すオプションを提供してください。	○	○	○			○	○	○	

2. 4　目的の提示とデータ保護方針

組織は、例外のない限り、個人に個人データ収集の目的を提示し、個人データを収集し、使用し、公開するための同意をとることが必要です。

組織が個人データの使用と保護だけでなく、データの収集、使用、開示などの目的やデータ保護方針を分かりやすく説明することに、ユーザの期待が高まっています。明確で簡潔な通知は、何よりも組織が優先して実行することのひとつです。

グッドプラクティス	SDLC フェーズ						3階層		
1. 収集の目的を説明 ユーザにとって個人情報の収集理由が明確でないときに特に必要になります。	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
個人データの収集に際して、項目ごとに必須な否かを明確にしてください。必須の項目では、ユーザが情報を提供しなかったときの結果を説明してください。		○	○			○	○		
2. 収集した個人データをリスト化 個人データについて「何の」情報を「どのように」収集したのかということをリスト化してください。例えば、GPS や Wi-Fi アクセスポイントを介して収集した位置情報という具合です。		○	○			○	○		
3. 第3者をリスト化 (もしあるのなら) 個人データを処理する第3者、どの個人データを第3者に提供するのか、そしてその目的は何かをリスト化してください。		○	○			○	○		
4. DPO の連絡先をリスト化 ユーザがデータ保護に関する質問をする際に、必要な DPO の連絡先がよく分かるようにしておいてください。		○	○			○	○		

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
5. つねにシンプル化 組織のデータ保護方針は簡潔にかつ読み易くするようにしてください。そうすることで、読まれる可能性が高くなります。		○	○			○	○		
6. 他のデータ保護方針を単純コピーしない データ保護方針案を作成する段階で、他の組織のデータ保護方針を参考にする際、自組織固有のニーズに合うように方針を修正することが重要です。すなわち新たな事項を追加したり、無関係な事項を削除したりすることです。		○	○			○	○		
7. 階層的にアプローチ データ保護方針にたくさんの情報が入りすぎるときには、まず概要を説明し、個々の詳しい内容はユーザが選択できるように検討してください。	○	○	○			○	○		
8. ジャストインタイムにアプローチ 情報過多を抑えるもう一つの方法は、ジャストインタイムあるいはダイナミックなアプローチ（個人データが収集される直前、もしくは許可を得る直前に通知）です。まずは短文で説明し、オプションとして全文を閲覧できるようにします。	○	○	○			○	○		
9. グラフィック情報を使用 文字ばかりのデータ保護方策で説明する代わり、いくつかの場面ではグラフィック情報で説明できるようにしてください。		○	○			○	○		

2．5　ユーザの個人データの同意を取得

ユーザの個人データを使用するためにユーザから同意を得る際に、ICT システムを効果的に役立てることができます。加えて、ICT によって組織が同意を管理し、同意記録を容易にチェックできるようになります。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1．ユーザが何に同意するのかを明確に説明 できるだけシンプルで簡潔な言葉を使ってください。もし文章が長くなる時は、概要をまず説明し、詳細な内容を説明するオプションを提供することを検討してください。		○	○	○		○	○		
2．合意はデフォルトのアクションではなく、ユーザに明示的なアクションを要求 ユーザが明示的な同意アクションを確実に行えるようにしてください。例えば、ユーザはチェックボックスにチェックマークをつけるようにします。（自動的にデフォルトが選択され）あらかじめチェックマークがついている方法を採用してはいけません。		○	○			○	○		
3．マーケティング資料を受け取る同意とは別に要求 マーケティング資料を受け取る同意をユーザから得るときには、例えば、2番目のチェックボックスを使うなど、個人データの同意とは別に要求してください。		○	○			○	○		
4．ユーザの同意内容を記録 ユーザがある項目に同意し、他の項目に同意しない場合があります。そのため、ユーザの同意内容や同意日時を記録に残すことが重要です。これは同意記録とも呼ばれます。	○	○	○			○	○	○	

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
5. 同意文面のコピーを保管 同意を得るために組織で使われる文面は時とともに変わる可能性があります。しかし、これらの文面は体系的にアーカイブされていないことが多いのです。そのため、異なるバージョンの文面のコピーを保管する適切な方法を考えてください。バージョンごとにそれが使われ始めた日付を保管しておくと、ある期間に使われた文面を確認することができます。バージョン管理ソフトによりアーカイブが実現できるでしょう。例えば、ユーザとの訴訟にその記録が役に立ちます。		○	○			○	○		
6. 合意の撤回を許可 ユーザに同意を撤回する手段を提供してください。手動による方法（例えば、メールで撤回を要求する方法）や自動的な方法によって実現できます。自動的な方法を使ったときには、同意撤回の記録を監査記録に反映させ易くなります。 ユーザが同意を撤回することがあれば、その結果をユーザに知らせることは有益です。また、組織にとっても同意撤回の要求状況をユーザに知らせることも有益です。	○	○	○			○	○	○	

2．6　ICT システムの開発

ICT プロジェクトを立ち上げる前に組織が直面するのは、新しいシステムを開発するのか、それとも既存システムを使用するのかという判断です。選択したアプローチによって、注意すべき事柄が変わってきます。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1．ICT ベンダーにセキュリティ要件を詳細に説明 ICT ベンダーにオーダーメイドのソリューションを開発してもらうときは、組織のデータ保護やセキュリティの要件をベンダーに詳細に説明することが必要です。それらの要件は開発作業の一部として文書化され、十分な要件であることを確認してください。	○	○	○	○	○	○	○	○	○
2．既製のソリューションのときは、使用する前にソリューションを理解 既製のソリューション（購入あるいはオープンソースのシステム）を使うときは、個人データを扱うソリューションが何をするのかを理解してください。個人データが十分に保護されることに、組織が確信を持てるときにだけソリューションの使用を開始することが重要です。 既製のソリューションあるいはコンポーネントを選択したとき、開発者によるサポートが充実しているか否かを検討してください。サポートのないソフトウェアは修復（パッチ）できない脆弱な部分、すなわち永続的な脆弱性が存在する可能性があります。	○			○		○			
3．パッチを適用 ICT システムの関連するコンポーネントに更新とセキュリティパッチをできるだけ早くテストして適用してください。これらをタイミングよく行うには、何らかの形で（手動あるいは自動）でモニタリングする必要があるでしょう。			○		○	○	○	○	○

2.7 オンライン方式

本節ではさまざまな形式のユーザが入力を行うウェブアプリケーションに関するガイドを述べます。主な脅威には、悪意のあるファイルのアップロード、クロスサイトスクリプティング（XSS）、SQL インジェクション、URL 操作があります。

詳しくは、PDPC の「電子メディアのための個人データの保護に関するガイドライン（Guide to Securing Personal Data in the Electronic Medium）」の中の「ウェブサイトとウエブアプリケーションセキュリティ（Websites and Web Application Security）」の章を参照してください。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1. HTTP でなく HTTPS を使用 HTTPS は、セキュリティ機能付きの HTTP で暗号化した情報を送ります。（セキュリティ機能なしの）HTTP では、組織のデータをそのまま送るため、ネットワークの途中で個人データが容易に抜き取られてしまいます。	○	○			○	○		○	
2. OWASP の最も重要なウェブアプリケーションリスク・トップ10[4] の対策を実装 このリストは、インジェクション、不適切なセキュリティ設定、クロスサイトスクリプティング、既知の脆弱性のあるコンポーネントの使用などの一般的にもっとも重要なウェブアプリケーションのセキュリティリスクについて提案しています。またそれらのリスクの防御のやり方についても説明しています。例えば、SQL インジェクションの防御には、クエリパラメータ化、エンコーディング、文字エンコード、文字エスケープ、入力検証があります。 OWASP トップ10プロアクティブコントロール[5] は、防御の技法と制御に焦点をあてた別の関連文書です。制御の各項目ではトップ10リスクのひとつ以上のリスクの防御策を説明しています。	○	○	○	○		○		○	

訳注：OWASP(オワスプ：The Open Web Application Security Project) https://www.owasp.org
注 4: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
注 5: https://www.owasp.org/index.php/OWASP_Proactive_Controls

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
3. ウェブアプリケーションファイアウォール（WAF）を使用 WAF は SQL インジェクションやクロスサイトスクリプティングのような典型的なウェブアプリケーション攻撃に対抗することが目的です。つまり、アプリケーションのコードレベルの対策に加えて、WAF はもうひとつのセキュリティ層として機能することができます。	○	○				○			
4. アップロードファイルをマルウェアスキャン ユーザがアップロードしたファイルには、故意か無意識かにかかわらず、マルウェアが入っているかもしれません。ファイルを何かに使う前には、ファイルにマルウェア対策が施されていることを確認してください。	○	○	○		○	○		○	
5. ユーザ入力を検証 ユーザが入力したデータが有効であることを確認するとともに、データ検証によって URL 操作や SQL インジェクションやバッファオーバーフロー攻撃などのセキュリティ侵害を防ぐことができます。	○	○	○			○		○	
6. 保存データを暗号化 セキュリティ強化のために保存する個人データを暗号化してください。保存データを暗号化する方法は： ・データベースで暗号化（アプリケーションでは処理しません） ・データをデータベースに保存する前にアプリケーションで暗号化する。データベースから引き出したデータはアプリケーションで復号化する必要があります。 鍵の管理が効果的な暗号化のために重要になります。 さらに、産業界が適切かつ安全な方法として認識していることを確認するために、定期的に暗号化の方法（例えば、アルゴリズムや鍵の長さなど）を見直してください。	○	○	○		○	○		○	○

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
7. より保護が必要な個人データを分離して保護 十分なセキュリティ機能による保護の強化が必要なときは、個人データの分離を検討してください。データを分離して付加的な保護を行う例として、以下のようなものがあります。 ・少数の選ばれたユーザグループのみにアクセス許可（いわゆる「知る必要性」原則（“need to know”basis）に基づく） ・高度な監視、警報、監査証跡、監査などの採用 ・より厳重なパスワードの要請（例えば、長いパスワードや頻繁なパスワードの変更） リレーショナルデータベースでは、一つの分離方法に水平分離があります。すなわちデータベースのレコードあるいは行でデータを分離します。もう一つの方法は、データベースのフィールドあるいは列の分離による垂直分離です。異なるデータベースに分けて保存する方法はより強固な保護になるかもしれません。また、同じデータベースでも異なるテーブルにするのがいいでしょう。 データ全体の量に比較してセンシティブな個人データが少数になるときには分離はより効果的になる可能性があります。	○	○	○			○		○	○
8. 自由記述欄の情報を注意して使用 ユーザが個人データを自由記述欄に常に書き込むことができることに注意してください。データ検証とともに、このような形式の個人データを検出するのは難しくはないでしょう。			○			○		○	

2．8　アクセス制御

ユーザが要求する個人データのアクセスを許可すべきか？認証され承認されたユーザだけが当該情報にアクセスを許可されていることを、組織がどのように確認できるのか？本節のグッドプラクティスはこれらの質問に対処するのに役立ちます。

詳しくは、PDPC の「電子メディアのための個人データの保護に関するガイドライン（Guide to Securing Personal Data in the Electronic Medium）」の中の「認証と承認とパスワード（Authentication, Authorisation and Passwords）」を参照してください。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1. 個人データの保存場所と流れを認識 保存場所が分からないと保護することができません。データベースの他に、個人データは一時ファイル、生成したレポート、他 ICT システムとの通信ファイル、バックアップ等にあります。これはシステムデザインの一部で、システムの設計者が、あるいはシステム仕様書が有用な情報を提供することが必要です。 データ項目や DPIA の詳細な情報は、本ガイドの前節を参照してください。	○	○				○			○
2. アプリケーションのアクセス制御を実装 セキュリティ技法であるアクセス制御は個人データを保護する基本的な方法です。これは多くの場合、認証（ユーザであることを確認）と承認（ユーザが要求されたリソースにアクセスする権限を持っているか否かを確認）のことです。 適切なアクセス制御の手段を使用してください。これは多くの場合、個人データの保存場所に依存します。例えば、アプリケーションからアクセスされる個人データの多くはデータベースに保存されており、その場合はアプリケーションでアクセス制御を実装します。 組織は、個人データへのアクセスについて、例えば、いくつかの属性だけはあるユーザのみがアクセスでき、他の属性は別のユーザだけがアクセスできるように、きめ細かく制御することを検討してください。	○	○	○			○		○	

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
3. アクセス制御をバイパスするバックドアの作成を回避 バックドアは、アプリケーションからアクセス制御を介さず個人データにアクセスする代替手段です。組織は、試験や一時的な目的でバックドアをつくることがあります。ただ残念なことに、バックドアを消し忘れたときは、ハッカーにとってはとても嬉しいニュースになります。ハッカーでなくとも、検索エンジンが見つけたバックドアからサーバーに入り、結果としてインターネットユーザが簡単に個人データにアクセスできるようになってしまいます。			○		○	○		○	
4. ICT システムの他の部分でもアクセス制御を設定 アプリケーション以外に、ウェブサーバやデータベースやファイルシステム、および ICT システムの他の部分でも、適切なアクセス制御を実装することが必要です。 例えば、構成の設定にもよりますが、ウェブサーバのファイルは、ファイルへのリンクが公開されていなくとも、そしてウェブサイトやウェブアプリケーションがリンクしていなくとも、インターネットから（誰でも）アクセスできるようになるかもしれません。ロボットを排除するプロトコル（robots.txt）を使用していても、検索エンジンによってそのようなウェブページや文書が見つからないとは保証できません。					○				○
5. ログイン失敗回数を制限 これはブルートフォース攻撃（総当たり攻撃）に対抗する手段です。以下のようなさまざまな仕組みを実装することができます。 ・X 回ログイン間違いするとアカウントをロック ・再ログインするまでの時間を長く ・ブルートフォース攻撃防止のためにキャプチャ（訳注：Captcha は Completely Automated Public Turing test to tell Computers and Humans Apart）を使用	○	○	○			○	○	○	

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
6. ワンタイムパスワード（OPT）あるいは多要素認証を使用 これはブルートフォース攻撃や他のハッキングに対抗する手段です。特に、管理者アカウントの保護やログイン後にアクセスできる個人データの機密性が高いと考えられる場合に有効です。	○	○	○			○	○	○	
7. パスワードの複雑さの規則 使用されるパスワードが業界推奨の複雑さの規則に準拠することを確認してください。例えば、パスワードの長さが8文字で、少なくとも1文字は大文字、1文字は数字、1文字は記号にするなど。 また、以前使っていたx個のパスワードの再利用を禁止することも検討してください。	○	○	○			○	○	○	
8. パスワードの保護 送信中（HTTPS のように暗号通信の使用により）や蓄積中（ハッシュ値のみでの記憶形態により）は、パスワードを保護してください。 定期的に暗号化の方法（アルゴリズムや鍵の長さなど）を見直し、適切で安全であることを業界に認識してもらうようにします。 画面でのパスワード表示は、アステリスクやドットのようなプレイスホルダー記号を見せるようにしてください。いくつかのアプリケーションでは、パスワード入力が安全に行われる（自分の他に画面を見る人がいない）ときに、パスワードそのものを見せるオプション機能を提供しています。	○	○	○			○	○	○	○
9. 定期的なパスワードの変更 パスワード変更の頻度を決める際には、個人データの機密性を考慮してください。また、ユーザに高頻度でパスワード変更を要求すると、「パスワード疲れ」になることもありますので注意してください。	○	○	○			○		○	

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
10. ユーザの役割あるいはユーザのグループを決めて適切なアクセス権限を付与 「最小権限(least privilege)」原則を使用し、できるだけ少ないアクセス権限を付与するように、ユーザに適切な役割を割り当ててください。ガイドラインとしては、ユーザは知る必要性のない情報を見ることができないように設定する必要があります。例えば、 ・人事(HR)部のユーザは、従業員の記録のみを閲覧できるが、顧客の記録を見ることができない。 ・人事部内では上級ユーザのみが従業員の機微情報を閲覧することができる。 これらはきめ細かな個人データへのアクセス制御の例です。	○	○	○			○		○	
11. ログインの成功と失敗の履歴を記録 ログイン失敗の記録はハッキング未遂の検出や調査に役立ちます。ハッキング検出には何らかの形でのログ監視が必要になります。	○	○	○			○		○	
12. ユーザアカウントを定期的に見直し すべてのユーザアカウントが正規のアカウントであることを保証することです。例えば、あるユーザが組織をやめると、そのユーザアカウントを更新あるいは削除する処理が必要になります。テストが終了したら、テストアカウントを削除することが必要です。 それとは別に、ユーザアカウントを定期的に見直すことも重要です。見直しでは、ユーザに付与された権限がほんとうにすべて必要なのか否かを確認することも重要です。						○			
13. 機微データへのアクセスを記録 組織では個人データへのアクセス記録、特に機密度の高いデータへのアクセスを記録することを検討してください。	○	○	○			○	○	○	

2．9　ICT システムの試験

アプリケーションが機能面で期待通りに動作することを確認するとともに、適切なリソースを考慮して、関連するセキュリティ試験を実施し、データ保護対策が意図したように動作していることを保証することが重要です。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1．試験環境では本番データの使用を回避 開発者は利便性のあることから、本番データ（よく個人データが含まれている）を使って試験を行う可能性があります。しかし、試験環境は本番に比べて安全性が低くなることが多く、個人データをリスクにさらすことになります。 組織は本番データの代わりに試験用のデータを新たに作ることを検討してください。 詳しくは、PDPC の基本データ匿名化技法のガイド（Guide to Basic Data Anonymisation Techniques）を参照してください。			○	○		○			○
2．SQL join を確認 長いステイトメントの SQL もあり、複雑な join 命令で構成されています。Join 命令に関するエラーは不明確なことがあり、結合される異なるデータサブジェクトに起因する可能性もあります。これによりデータ漏えいになりかねません。		○	○	○		○		○	
3．コードを見直し 大きな影響があると評価されたセクションのソースコードは少なくともコードの見直しが必要です。もし、手作業で行うのであれば、経験豊かな開発者に見直しをしてもらうのが賢明です。			○	○		○	○	○	○

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
4. 脆弱性を評価 脆弱性が既知であり、新たなパッチを適用していないコンポーネントなど、何らかの脆弱性をシステムが含んでいるか否かを確認してください。これは通常、ソフトウェアツールを用いて行います。				○	○	○	○	○	○
5. 侵入試験を実施 侵入試験には特別な技能が必要ですが、組織は必要なら外部の関係者に支援してもらうことができます。この目的は ICT システムにどのように侵入するかを見極め、システムの脆弱性を見つけて修正することです。				○		○	○	○	○
6. ユーザ受容試験（UAT）を実施 システムの機能性を検証するとともに、システムが提示するデータ保護対策の使いやすさ、データ保護指針や実践の分かりやすさを検証するために、UAT を実施することも必要です。				○	○	○	○	○	○

2. 10 ICT システムにおける個人データのアクセス、訂正、正確さ

個人情報保護法（PDPA；Personal Data Protection Act）との整合性とともに、アクセスや修正に関する組織の義務についても、ICT システムを役立てることができます。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1. ユーザに自己管理機能を提供 可能であれば、従業員が援助せずにユーザが個人データを管理できるようにしてください。これによりヒューマンエラーの発生が最小限になるとともに、従業員の手間と時間を節約できます。 例えば、これはポータル（訳注：サイトの入口）で実行することもできます。なお、ユーザが自分の個人データにアクセスする前、ポータルでユーザにログインしてもらうことが重要です。	○	○	○			○	○	○	
2. ユーザによる更新の確認 次のステップはユーザが自己管理機能を確実に使うようにしてもらうことです。これにはいろいろな方法があります。例えば、定期的にユーザに使ってもらうように仕向けたり、詳細な情報を見せて正しいことを確認してもらったりします。	○	○	○			○	○	○	

2. 11　ICT システムにおける個人データの管理

組織は主に個人データのメイン記憶（例えば、そのデータベース）の保護に力を入れるのですが、一時的なあるいは1回だけ使われる個人データの 2 次記憶媒体の保護を省略してしまうこともあります。そのため、システムの管理（ハウスキープ）は特に 2 次記憶に関連しています。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1. 個人データを含む一時ファイルが恒久的になることを防止 一時ファイルが生成されることがあります。例えば、他システムとやりとりするための中間ファイルなどです。この一時ファイルが存在している間はそのファイルを保護し、不要になったら削除するようにしてください。ハウスキープが自動的に削除を行うようにスケジュールしておいてください。		○	○		○	○		○	○
2. データの移行ファイルに注意 移行ファイルは個人データがより脆弱な形式になるために特に注意が必要です。例えば、（通常は）セキュリティ層で保護されているデータベースのファイル形式の代わりに、CSV 形式のファイルを用いるときなどです。また、移行期間が長引き、結果として長期間にわたって脆弱性が生じるというリスクもあります。移行が終わったら移行ファイルの存在を忘れてしまい、恒久的な脆弱性につながってしまいます。		○	○		○	○			○

2. 12 ユーザデバイスのセキュリティ

従業員が使うコンピュータデバイスのセキュリティはますます重要になってきました。というのはラップトップやタブレットのようにコンピュータデバイスをどこにでも持ち歩くのが一般的になってきたからです。(従業員がどこにでも持参する)組織のコンピュータデバイスは、組織の ICT システムやそこに入っている個人データへの入口になってしまいます。

詳しくは、PDPC の「電子メディアのための個人データの保護に関するガイドライン(Guide to Securing Personal Data in the Electronic Medium)」の中の「ポータブルコンピュータデバイスとリムーバブルな記憶メディア(Portable Computing Devices & Removable Storage Media)」の節を参照してください。

グッドプラクティス	SDLC フェーズ						3階層		
1. 従業員のコンピュータデバイスを保護 検討にあたいするグッドプラクティスは以下の通りです。	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
・デバイス使用を管理する方針を策定 ・モバイルデバイスマネジメントを採用(例えば、遠隔データ消去) ・デバイス上のデータを暗号化 ・データのセキュアな消去を実施 ・マルウェア対策のソフトウェアを使用 ・デバイス上の個人データ保存を最小化 ・強制的なデバイスへのログインを実施 ・不使用中にスクリーンロックを使用				○	○	○			

2. 13 データのエクスポート

データを単独のファイル（例えば、PDF フォーマットの報告書）に移行したり、他の ICT システムにデータを転送したりするデータのエクスポートがよくあります。データが一度エクスポートされると「オフライン」の状態になってしまい、もともとのアプリケーションに具備したアクセス制御ではデータの保護ができなくなります。そのためエクスポートされた形態でもデータを保護する必要があります。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1. エクスポートするデータを暗号化 さまざまな暗号のアルゴリズムを使うことができます。暗号化の強さはアルゴリズムや暗号鍵の長さによって異なります。 大事なことは、暗号鍵の安全を確保し暗号化したデータとは別にするなど、暗号鍵を適切に管理し保護することです。 暗号化の方法（アルゴリズムや暗号鍵の長さ）を定期的に見直し、それが適切で安全であることを業界に認識してもらうようにします。	○	○	○			○		○	
2. 暗号鍵とは別に送信 エクスポートしたファイルを受け取った相手にその暗号鍵を送る必要があるときは、ファイルとは別に鍵を送信してください。例えば、もし電子メールで暗号化したファイルを送るときは、暗号鍵はファイル送信とは別のメールで送るようにしてください。あるいは、もっと良い方法は別の通信方法を使うことです。	○	○	○		○	○		○	

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
3. 匿名化の技法を適用 オリジナルのデータの代わりに、匿名化したデータを送ることで十分な場合もあります。匿名化の技法には、文字のマスキング、仮名化、一般化、データの集約などがあります。 例えば、ある組織では、GIRO（訳注：シンガポールのインターネット送金のひとつ）控除に使うために、月次計算書で部分的にマスクした口座番号をリスト化します。 詳しくは、PDPC の基本データ匿名化技法（Guide to Basic Data Anonymisation Techniques）を参照してください。	○	○	○			○	○	○	
4. 新たなプライバシー保護技術を活用 個人データの共有や活用を行うとともに、保護も行う新しい技術が登場している。例えば、差分プライバシー（differential privacy）、準同型暗号（Homomorphic Encryption）、秘密計算（secure multi-party computation）などがあります。ただ、このガイドを発行した時点では、いくつかの技術だけがソフトウェア製品の形として使われ始めた段階です。	○	○	○			○	○	○	

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
5. 電子メールの送付 個人データを含む電子メールを送るアプリケーションを使う前には、組織は徹底した試験を行う必要があります。というのも、自動送信では瞬く間に大量の電子メールが送られてしまい、個人データが間違った相手の送られたことを気が付いても、メールの送信を止めたり、送られたメールを回収したりすることが困難だからです。 同じメールを複数の相手に送るとき、特に何か機密性のある場合は、相手先を BCC にして送るようにしてください。 電子メールで直接個人データを送る代わりに、ユーザが情報を見る際に認証を必要とするリンク先を送ることもできます。	○	○	○	○		○		○	
6. データのエクスポートを監視 データのエクスポートが可能な限度を設定することを検討してください。またデータの不正流出を摘発するためにデータのエクスポートを監視してください。					○				○

2. 14 ICT システムにおける個人データの保存

電子的なデータ形式では、ICT システムを使うことで保存期間が終了になった個人データの検出をより容易に組織で実現できるようになります。これを実現するには、最初からシステムの設計要素に組み込んで込んでおくことが必須になります。

グッドプラクティス	SDLC フェーズ						3階層		
1. 期限切れの個人データを適切に管理 ICT システムにより保存期間の終了したレコードにフラグを立てることができます。これらのレコードを削除する、もしくは組織のポリシーにしたがって処理（例えば、匿名化）することが必要です。	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
	○	○	○			○	○	○	○

2. 15 保守フェーズ

ICT システムが稼働し安定した後、技術チームは人員やその他のリソースに関して縮小されるのが一般的です。しかし、個人データ保護については十分なリソースを確保することが大切です。このフェーズにおけるシステムの機能強化は、「ミニ SDLC」としてとらえることができます。つまり、保守フェーズでの機能強化は保守よりも前のフェーズで実施されてきたことを見直すことなのです。

グッドプラクティス	SDLC フェーズ						3階層		
	要件	設計	開発	試験	移行	保守	表現	アプリ	データ
1. 新しい法律あるいは法律改正に注意 法律は時とともに変わります。そのため個人データに関する要求や義務も変わります。						○	○	○	○
2. 技術の変化に注意 同様に、技術の変化によってシステムの設計や実装で調整が必要になる可能性があります。例えば、現在安全な暗号化の標準でも、将来ずっと安全でグッドプラクティスであるとは限りません。						○	○	○	○
3. データ目録の更新 システムに変更が生じたときに、個人データの目録を更新してください。						○	○	○	○
4. 定期的にセキュリティ試験を継続 システムが動いている限り、セキュリティ試験に終わりはありません。たとえシステムに変更が生じなくとも、新たな脆弱性が出現することがあります。システム構成の変化も、結果として脆弱性を生むことになります。						○	○	○	○

謝 辞

本ガイドの作成に関して貴重なご意見を頂いた以下の組織に感謝の意を表します。

AsiaDPO

Cyber Security Agency of Singapore (CSA)

Government Technology Agency (GovTech)

The Law Society of Singapore

SGTech

Singapore Computer Society (SCS)

The Software Alliance (BSA)

本ガイドラインは以下の組織により作成された。

Personal Data Protection Commission (PDPC), Singapore
Privacy Commissioner for Personal Data (PCPD), Hong Kong