1.設計書には以下のような種類があります。それぞれの役割は次の通りです：

* 要件定義書: お客様の要件やシステムの目的を文書化し、業務内容や機能要件を説明します。これはシステムの基本的な設計指針を提供します。
* 外部設計書: 要件定義書をもとに、実際のシステムの外部インターフェースや画面デザイン、入力項目、ボタン、画面遷移、入力チェックなどの詳細な設計を行います。ユーザーインターフェースの設計に焦点が当てられます。
* 内部設計書: システムの内部処理やデータベースの設計、サーバとの連携など、システムの実装に必要な詳細な設計を文書化します。プログラムの仕様を提供し、開発者が実際のコードを書くためのガイドとなります。
* DB設計書: データベースの設計に関する情報を提供します。データベースのテーブル構造、関連性、インデックス、データ型などが含まれます。
* API設計書: システムが提供するAPI（アプリケーションプログラミングインターフェース）に関する情報を提供します。APIエンドポイント、リクエストとレスポンスの形式、認証方法などが含まれます。

2. 設計書作成後のレビューの主な目的は次の通りです：

* 要件の確認: 要件定義書に記載された要件が適切かつ完全であることを確認します。不足している要件や不明瞭な点を特定します。
* 設計の一貫性: 要件定義書と外部設計書、内部設計書などの間で一貫性があり、設計が要件を満たすことを確認します。
* 問題の特定: 設計に関する問題や誤りを特定し、修正が必要な部分を明らかにします。

3. 設計書を読むことで理解できることは次の通りです：

* システムの機能や業務内容
* ユーザーインターフェースのデザインと操作方法
* データベースの構造とデータの保存方法
* システム内部の処理フローとロジック
* システム間のデータや通信フロー

4. 設計書が日本語で書かれている場合、以下の対応策が考えられます：

* 日本語スキル向上: 日本語の理解力を高めるために、日本語学習を積極的に行います。
* 翻訳ツールの使用: オンライン翻訳ツールを利用して、設計書を自分の言語に翻訳して理解するのに役立てることができます。

5. 設計書の読解に苦労したり、期日までに読み終えるのが難しい場合、以下の対策を検討することができます：

* スケジュール調整: プロジェクトリーダーや上司と相談し、期日の延期をリクエストするか、読解の補助を得るためのサポートを依頼します。
* 分割読解: 設計書を段階的に読むことで、大量の情報を少しずつ処理します。毎日少しずつ読むことで、期日までに読み終えることができるかもしれません。
* 質問とクラリフィケーション: 不明瞭な部分や疑問点があれば、設計担当者に質問し、理解を深めます。