

上流工程革新構造

本質はシステムの構造原理にあった

LEXAR Corporation | contact@lxr-j.com

上流工程の問題点

上流工程の問題点について、客観的な意見を得るため Chatgpt に聞いてみました。以下のように 10 個の問題点が提示されました。LEXAR では、このような問題は全てシステムの原理が生むと考えています



システム開発の上流工程（企画、要求定義、システム設計など）における主な問題点は以下の通りです。

1. 要求定義の曖昧さ:
 - クライアントやエンドユーザーからの要求が十分に明確でない場合、システムの目的や必要な機能が正確に定義されず、後の工程で手戻りが発生する可能性が高くなります。要求が抽象的であると、開発チームとクライアントの間で認識のズレが生じ、最終的に期待通りのシステムが出来上がらないリスクがあります。
2. ステークホルダー間のコミュニケーション不足:
 - システム開発プロジェクトには多くのステークホルダーが関わりますが、彼ら間でのコミュニケーションが不足していると、要求や期待が適切に共有されないことがあります。これにより、重要な要求が見落とされる、または誤解される可能性があります。
3. 要件の優先順位付けが不十分:
 - 要求定義段階で、全ての要求に同等の重要性が与えられた場合、プロジェクトの進行中にリソースの分配が非効率になり、最も重要な機能や要件が適切に実現されないリスクがあります。優先順位付けが不十分だと、プロジェクトの終盤で重要な要件に対するリソースが不足する可能性があります。
4. 変化する要求への対応の難しさ:
 - プロジェクトが進行する中で、ビジネス環境やユーザーのニーズが変化することがありますが、これに柔軟に対応できる仕組みがないと、後の工程で大きな手戻りが発生する可能性があります。上流工程での要求の凍結が厳格すぎる場合、プロジェクトの柔軟性が失われ、結果としてシステムが時代遅れになるリスクがあります。
5. スコープクリープの発生:
 - プロジェクトの範囲が明確に定義されていない場合、要求や機能が追加され続け、最終的にプロジェクトの範囲が大幅に拡大してしまうことがあります。このスコープクリープにより、予算超過やスケジュールの遅延が発生しやすくなります。
6. 技術的制約の考慮不足:
 - 要求定義段階で技術的な制約や現実的な実装可能性が十分に考慮されていない場合、後の設計や開発フェーズで重大な問題が発生する可能性があります。これにより、要求を満たすための開発が困難になる、または追加コストが発生するリスクがあります。
7. 業務プロセスの理解不足:
 - システムがサポートする業務プロセスやビジネス要件が十分に理解されていない場合、システムの設計が実際の業務に合わない可能性があります。これにより、システムが導入後にユーザーに受け入れられない、または業務効率を低下させるリスクがあります。

8. 要件のトレーサビリティ不足:

- 要求定義から設計、実装、テストに至るまでのプロセスで、各要件がどのように実現されているかを追跡する仕組みが不足していると、要件が設計や実装で適切に反映されないリスクがあります。これにより、システムが最終的に要求を満たさない可能性があります。

9. コスト見積もりの不正確さ:

- 上流工程でのコスト見積もりが不正確だと、プロジェクト全体の予算が適切に設定されず、プロジェクト後半で資金不足に陥るリスクがあります。これにより、機能の削減や品質の低下が発生する可能性があります。

10. スケジュール管理の不備:

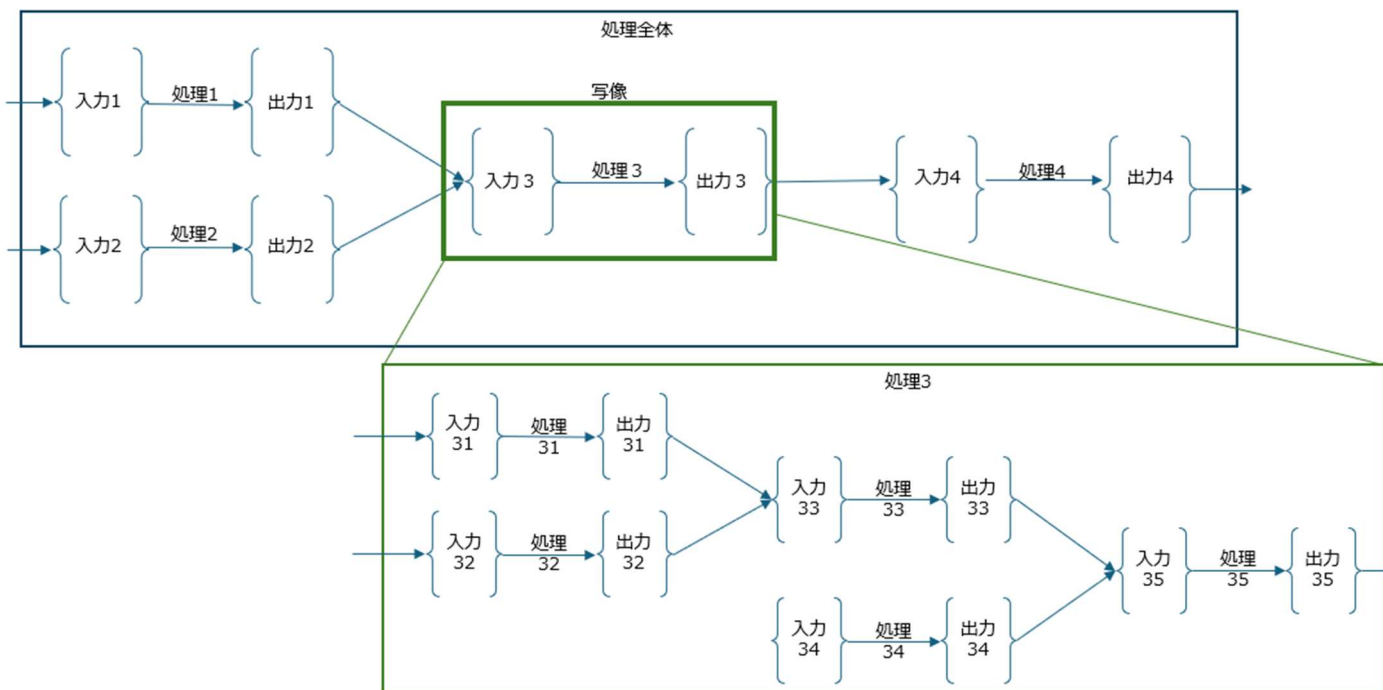
- プロジェクトのスケジュールが現実的でない場合や、各工程の作業量が適切に見積もられていない場合、プロジェクトが遅延し、納期に間に合わないリスクが高まります。特に、上流工程での遅延は後続の工程に大きな影響を与えるため、全体のプロジェクトが遅れる原因となります。

問題の因数分解方法

なぜ先述したような問題が発生するのかを考察する際に、事象を因数分解していく必要性があります。因数分解するためのフレームワークとして、非常に当たり前ですが、写像を用います。写像と難しく言いましたが、

$$Y = F(X)$$

つまり、入力 X を処理 F して出力 F という論理の基本であり、中学生で習うものです。論理的なものはすべてこの写像がなりたっていますから、業務やシステムももちろん、適用できるはずで

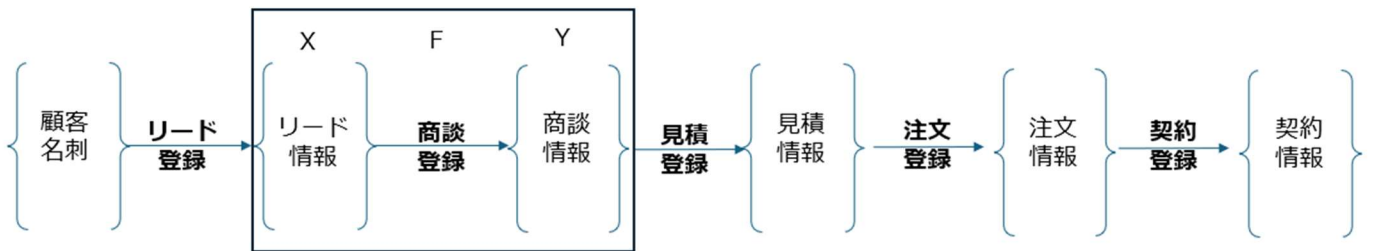


システムの写像構造

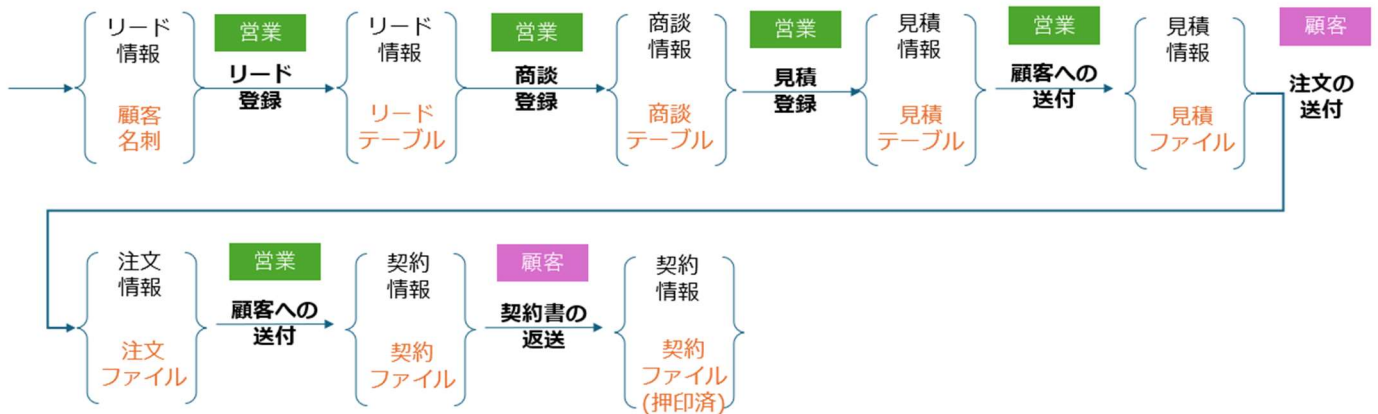
では、簡単な販売管理業務に適用してみるとどうでしょう。下図のように入力に合わせて、処理があって、出力となり、出力と次の入力はイコールとなっています。まさに、 $Y = F(X)$ の形態になっていますね。

ただ、実態はもう少し複雑ですので、見ていきましょう。

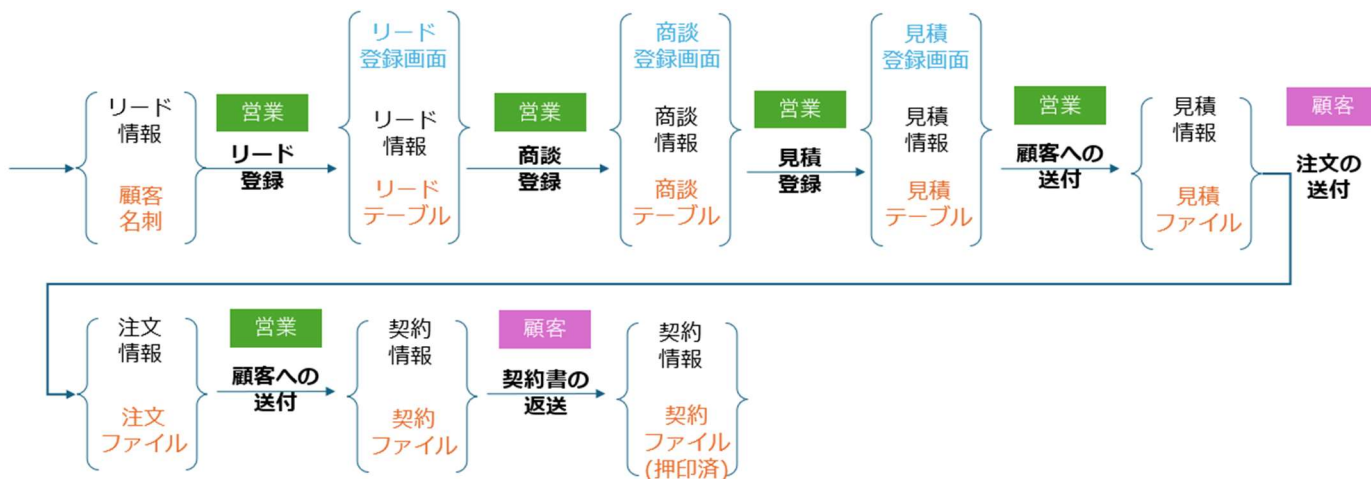
その前に、以降の議論をしやすくするため、今後 X や Y を構成、F を処理と呼びますので、覚えておいてください。



下図は、上図を論理データと物理データに分解し、ロールの概念を入れてみました。一番最初の論理データのリード情報は、同じ論理データでも顧客名刺とリードファイルという異なる物理データと紐付いています。このように、物理データは論理データを置く皿のようなものですので、料理という論理データが同じでも皿だけ変わることがあります。また、物理データには基本的に対応する論理データが存在しますので、物理データという構成と論理データという構成は関連付けが必要です。以降、これを構成条件と呼びます。

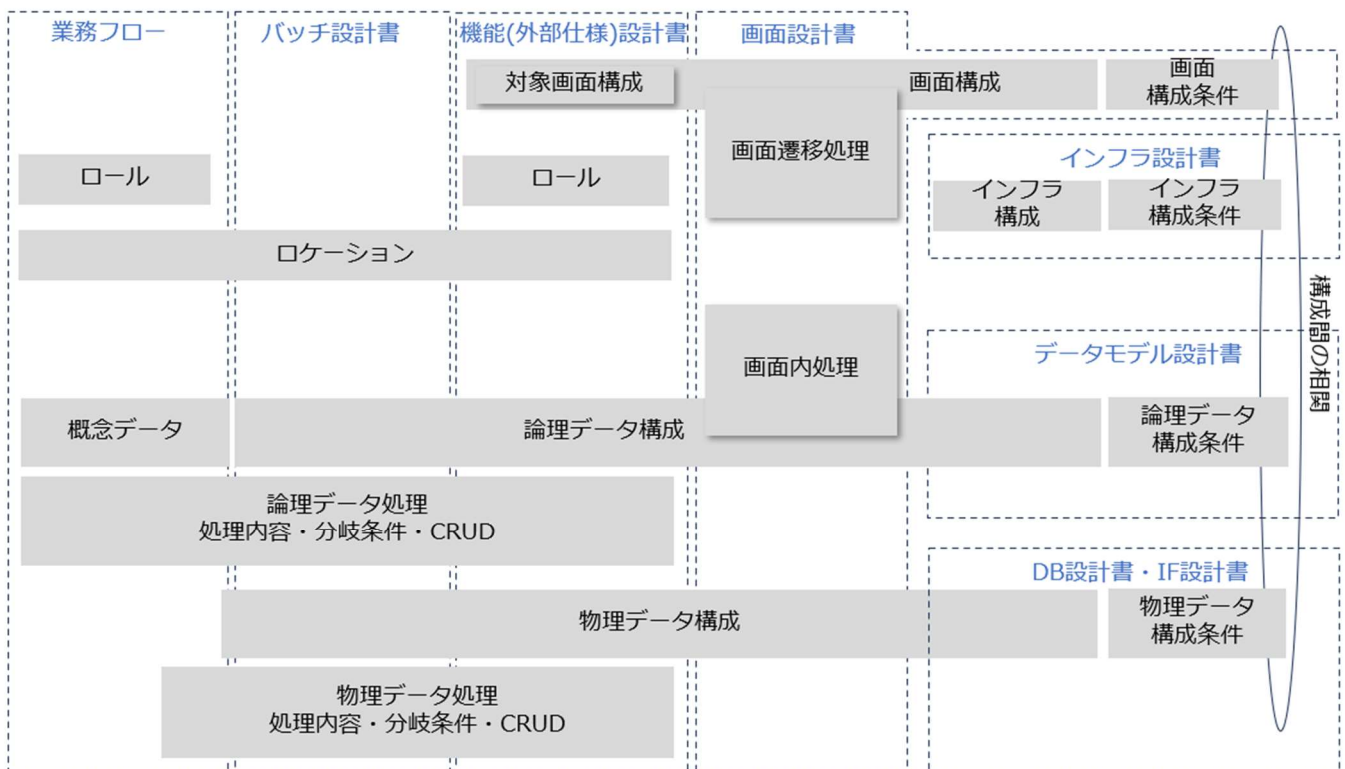


更にこの物理データと論理データの関係に加えて、登録する画面について情報を加えたのが以下です。このように、画面も構成ですから、3つの構成は関連していることがわかります。



従来型設計書の情報分断構造

システムは構成と処理から成り立っており、構成と構成は構成条件で関連付いていると前項までで整理しました。では、従来型の設計書と構成と処理の関係はどうなっているのでしょうか。下図はその関係を図式化したものです。これまでは、各設計書は個別に作成され、構成や処理の跨りは人間が保証していました。またツールを使う場合もツールがバラバラだと、構成や処理の跨りの問題は解消しません。



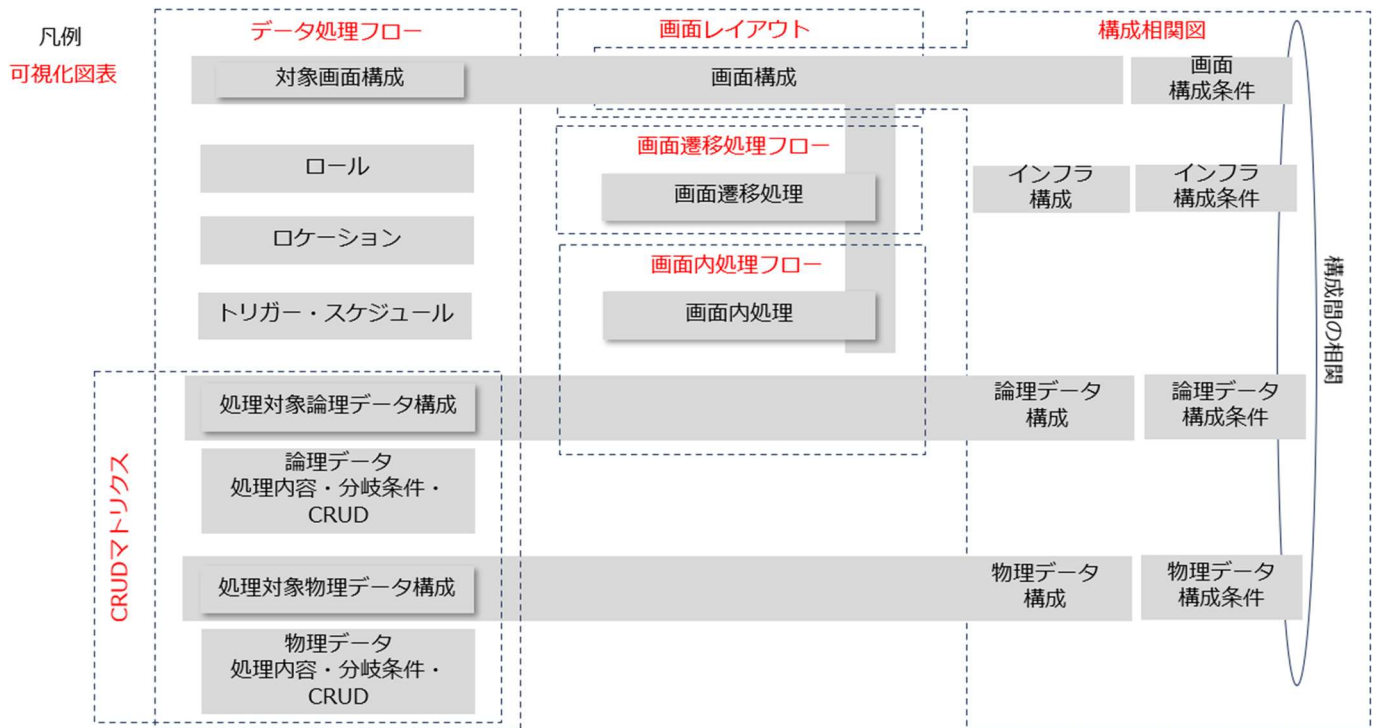
このように設計書間をまたがる構成や処理には下図のように障壁が存在します。この障壁により設計書と設計書の境界は分断されます。この分断こそが、システムが持つ原理からくる問題点です。複雑化・肥大化したシステムはより多くの障壁を作り出します。この障壁の多さが人間の限界を超えていることからブラックボックスは起きています。



XERV による情報構造化

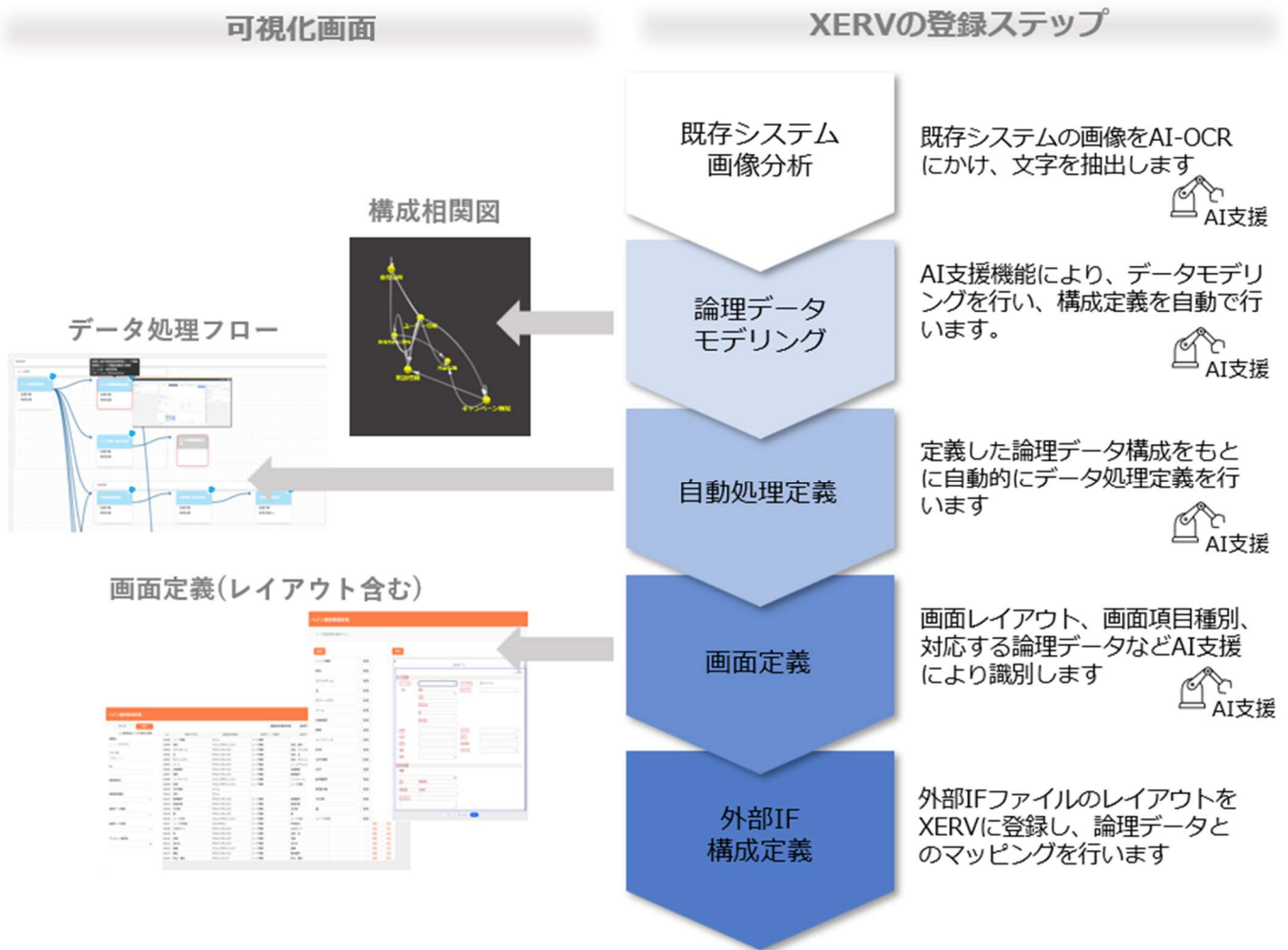
このような情報の分断を起こさないようにするには、全ての構成や処理を構造化して写像化する必要があります。これまでの、この構成と処理の写像が出来なかったので一連の設計(仕様)を統合的に管理出来ませんでした。株式会社 LEXAR は長年の研究成果としてこの構成と処理の写像に世界で初めて成功しました。

これにより構成と処理は、以下のようにシームレスに構造化され、構造化を実現したことで自動的な可視化も可能になります。その証拠として、所謂業務フローとシステムフローを統合したデータ処理フローは構造化されたデータから自動描画されます。また構成が構造化されると、例えば既存システムの仕様書が無い場合でも画面を OCR で読み込んでデータモデリングも自動化して仕様を定義することも可能になります。XERV ではこのような AI の活用による登録自動化も実現しています。



XERVによる上流工程の進め方

XERVによる上流工程の進め方はこれまでと若干異なります。AI支援マークはAIなどで自動化支援されるものです。ご覧のように殆どのステップで自動化支援が機能します。このように多くのAIによる登録支援機能を活用して構成や処理を登録することで、データ処理フローや構成相関図など可視化機能をより少ない工数で利用することが出来ます。



XERVによる課題解決

冒頭でご紹介した上流工程の問題点について XERV を用いるとどのように解決するのか同じく Chatgpt に聞いてみました。その結果が以下です。このように XERV による仕様管理を行うことで上流工程のあらゆる課題を解決することが出来ます。



以下に、システム開発の上流工程における10の問題点がXERVでどのように改善されるかを説明します。

1. 要求定義の曖昧さの解消:

- XERVは要求を視覚的に整理し、ステークホルダー全員が理解しやすい形で共有します。これにより、要求が不明確な場合でも、全員が同じ理解を持つことをサポートします。要求変更がプロジェクト全体に与える影響を迅速に把握し、適切な対応を促進します。

2. ステークホルダー間のコミュニケーション強化:

- XERVはシステム全体の設計や要求を一元的に可視化し、ステークホルダー間の情報共有を促進します。これにより、全員が同じ目標や要求を理解しやすくなり、コミュニケーションの欠如から生じる問題を減らすことができます。

3. 要件の優先順位付けをサポート:

- XERVは要件の重要性や依存関係を視覚的に整理し、プロジェクトにおける優先順位付けをサポートします。これにより、最も重要な要件にリソースを集中させることができ、プロジェクト全体の効率を向上させます。

4. 変化する要求への柔軟な対応:

- XERVは、要求や設計の変更がプロジェクト全体にどのような影響を与えるかを可視化することで、変化するビジネスニーズやユーザーの要求に柔軟に対応できるようにします。これにより、手戻りや再設計のリスクを最小限に抑えた形でプロジェクトを進行できます。

5. スコープクリープの防止:

- XERVはプロジェクトの範囲を明確に定義し、プロジェクト管理者がスコープクリープを防ぐためのツールを提供します。スコープ外の要求や機能が追加されるリスクを早期に検出し、プロジェクトの進行に影響を与えないように管理できます。

6. 技術的制約の考慮:

- XERVは、技術的な要件や制約を視覚的に整理し、それらが要求や設計に与える影響を明確にすることができます。これにより、技術的制約に基づいた現実的な設計と開発が進められるようになります。技術的なリスクや制約を初期段階で把握し、対応策を講じることができます。

7. 業務プロセスの理解を深める:

- XERVは業務プロセスとシステム要件を統合的に可視化し、業務に即したシステム設計を支援します。これにより、システムが実際の業務プロセスに適合するように設計され、導入後にユーザーが使いやすいシステムを構築できます。

8. 要件のトレーサビリティ確保:

- XERVは、要求定義から設計、実装、テストに至るまでのプロセスで要件のトレーサビリティを確保します。これにより、各要件がどのように実現されているかを容易に追跡でき、システムが最終的に要求を満たすことを保証します。

9. 正確なコスト見積もりの支援:

- XERVは、要求や設計の詳細に基づいてコスト見積もりをサポートします。これにより、プロジェクト全体の予算が現実的に設定され、後の段階での資金不足リスクを低減できます。

10. スケジュール管理の最適化:

- XERVはプロジェクトのスケジュールを可視化し、各工程の進捗やリスクを早期に検出します。これにより、スケジュールの最適化が可能となり、納期に間に合うようにプロジェクトの進行を調整することができます。