CADプロジェクトにおける効率化の検討と実践

宮城大学大学院　事業構想学研究科
佐藤弘和

本日の発表

・研究目的

・実際の現場での経験による検討と対応後の改善の効果について

・システム開発による開発方法論

・アジャイル方法論に基づくCASE.1～3の改善案

研究目的

・ウォーターフォール方式のハードウェア開発における問題点と、
検討・実践した際の効果について、実体験より述べる。

・CADプロジェクトにおける情報共有の効率化、仕様変更への対応、
手戻しによる損失の最小化について、情報システム開発における
アジャイルなプロジェクト管理と対比しながら論じる。

CASE.1 液晶ディスプレイにおけるＣＡＤ設計

＜内容＞

・携帯電話、PDAの液晶ディスプレイ設計

＜担当業務＞

・ウェハ上の最大取り数の検討、フォトマスク設計

・画面表示・機能実装デバイスの設計

・ディスプレイ内の配線設計

図1　携帯電話

図2　PDA端末

＜フォトマスクとウェハ上の取り数検討＞

・フォトマスク

→ディスプレイを機能させる部品や、それを制御する回路を製造するためのパターンをウェハ上に作成するため露光装置と組み合わせて使用する遮光板

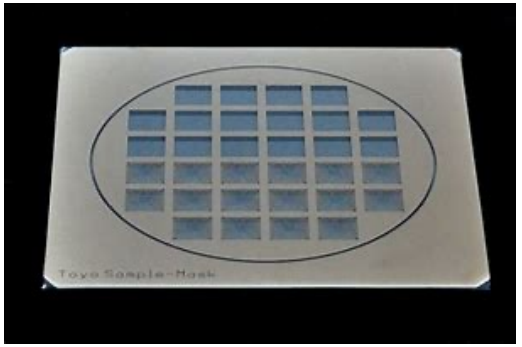


図3　フォトマスク
（□15cm程度）

①ウェハ上に溶剤Ａを一様に塗布する。
②フォトマスクＡを取り付けた露光装置でウェハを露光する。
③光が当たった部分の溶剤Ａが固まって、ウェハに固着される。
④洗浄液でウェハを洗い流す。パターンＡが作成される。
⑤パターンＡが作成されたウェハ上に溶剤Ｂを一様に塗布する。
⑥フォトマスクＢを取り付けた露光装置でウェハを露光する。
以上の工程を繰り返し、必要なパターンを形成していく。

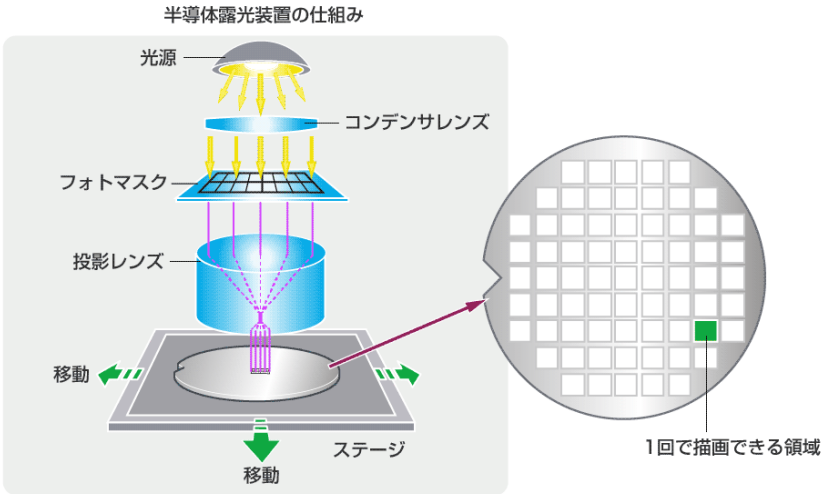


図4　半導体露光装置の仕組み

＜フォトマスクとウェハ上の取り数検討＞

・ウェハ上の取り数検討
　→既定サイズのウェハに、製造ルールに従ってディスプレイが
　　最大取り数となる配置の検討

・ウェハ1枚に対し数多くディスプレイを作成出来るほど
　１枚当たりのコストが下る。
・フォトマスクの枚数や露光回数を減らすほどコストが下る。

設計者は製造ルールと最大取り数との兼ね合いを見て、
いくつか候補となる配置案を出し、液晶メーカーに提出

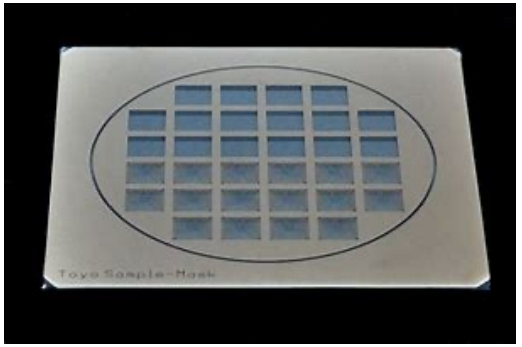


図3　フォトマスク
（□15cm程度）

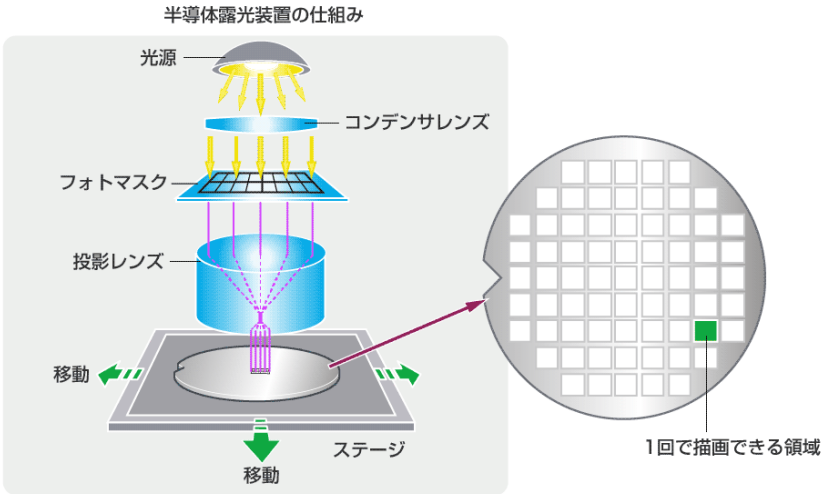


図4　半導体露光装置の仕組み

＜CASE.1における問題点＞

・最終製品メーカー⇔液晶メーカー⇔設計会社(自身が所属)という連絡系統となるため、

情報要求に対するレスポンスが悪かった。

→仕様の不明瞭・ルール上不可能な場合などに迅速に回答を得られない

→納期に対応するため、想定されるパターンで見切り発車のような形で対応することも多く、結果として無駄な作業や確認不十分によるミスが発生したり等、作業効率を悪くする一端となっていた。

＜CASE.1における改善案＞

問題点を纏めると以下の通りである。

・情報のレスポンスが悪く、無駄作業やミスが増える。
**　結果として作業効率が落ちる。**

この問題の改善案については以下の２つになる。

（１）　情報伝達速度の向上
（２）　設計作業の自動化

（１）については問題の本質的なところなので、第一に改善に着手されるべきところである。

（２）について、液晶の設計はルールや配置スペースによる制約が大きく、それ故作業を標準化できる部分も多い。従って、標準化できる部分についてプログラムによる自動化を行うことで作業効率を上げ、対応に余裕を持とうというアプローチである。

＜CASE.1における実践＞
自分を含め何名かで液晶メーカー社に出向した際、情報伝達速度の改善について
液晶メーカーと設計会社で話し合いの場が持たれた。
結果、最終製品メーカー⇔液晶メーカー⇔設計会社という連絡系統は変えようがなく、
早期の改善は難しいとのことだった。

作業自動化については、当時使用していたＣＡＤツール上で動くマクロ、および添付
する資料のExcel関数とＶＢＡを用いた入力フォーム化に取り組んだ。
その結果、仕様上の数値を入力することで最大取り数の検討を行う際の初期ＣＡＤデータ
と添付するExcel資料の作成自動化に成功し、作業効率はかなり向上した。
(結果として、自動化ツールありきの納期に短縮されてしまったのは皮肉である)

＜CASE.1における今後の課題＞

本項における問題点は多重下請け構造における下請けの弊害であり、組織構造の変更は容易ではない。特にハードウェア業界は製品が形になるまで期間がかかり、短いサイクルでウォーターフォールの工程をこなすというのが難しいため、ソフトウェア業界と比較しアジャイル方式の導入が遅れている要因になっているのではないかと考察される。

CASE.2 ダンプ艤装におけるCAD設計

＜内容＞

・特殊ダンプ・スライドデッキの艤装設計

・モデルチェンジ対応

＜担当業務＞

・艤装品の設計

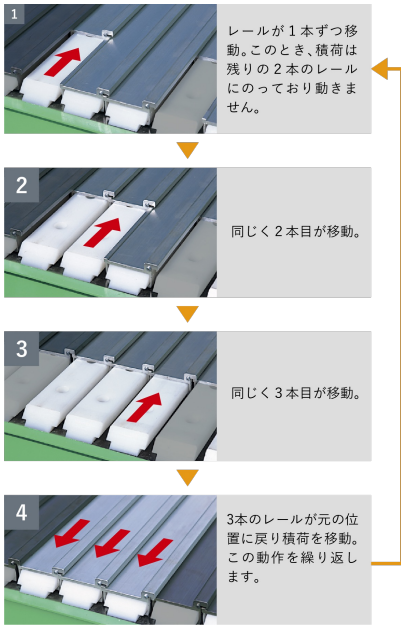
・艤装品と車両本体との取付検討と図面作成

・顧客・製造部署との調整

・モデルチェンジによる規格変更対応

図5　特殊ダンプ

図6　スライドデッキ



＜CASE.2における問題点＞

・納車されてくる本体車両ごとに、艤装品を取り付けるために開けられた穴や溶接ビス等の取付禁止箇所(本体と艤装品が干渉して取り付けられない)の位置と寸法のばらつきが大きい。

→生産工程での混乱を避けるため、図面は各案件で共通に使えるものでなければならず、検討に非常に時間が掛かる

（例えば、３台生産する案件だった場合、３台すべての製造に対応できる図面を作成しなければならなかった。）

・実際に図面作成を行うに当たっては、納品された実車を手作業ですべて実測し、数値をCADに落とし込んで全車共通で艤装品が配置可能な個所を割り出した上で、そこから正式な図面に仕上げるといった手順が必要であった。

・さらに、実測値を使用して製図しているため、他の設計者にクロスチェックを依頼してもチェックの精度があまり保証されないという問題があった。

＜CASE.2における改善案＞

問題点を纏めると以下の通りである。

・本体車両ごとの個体差が大きいために設計の自由度が下がり、図面作成の前段階での検討時間が大きくとられてしまう。

・クロスチェックの精度が下がりミスが増えてしまう。

この問題の改善案については以下の２つになる。

（１） 車体ごとのばらつきを無くしてもらう

（２） 車体の数値を測定する際に他設計者に同行してもらう

＜CASE.2における実践＞

（１）について、本体車両は車体メーカーから納車されてくるもののため、こちらで改善できるものではない。自身の所属会社としても現状の状態で作業が回っていたため特に指摘されることもなく、自身が契約満了で退社するまで改善されることはなかった。

設計側でやれることとしては、（２）の本体車両の数値を測定する際に他設計者に同行してもらいクロスチェックを担当してもらうことで測定時のミスを減らし、精度を上げるようにしていた。ただし、ミスの頻度は下がったものの、それ以上に各作業者の作業量が増えてしまったため、有効な改善点とは言えなかった。

CASE.3バス艤装におけるCAD設計

＜内容＞

・路線バスの艤装設計

・モデルチェンジ対応

＜担当業務＞

・艤装品の設計

・艤装品と車両本体との取付検討と図面作成

・電装品の取付位置検討と図面作成

・顧客・製造部署との調整

・モデルチェンジによる規格変更対応

図7　路線バス

図8　運転席周り

＜CASE.3における問題点＞

・納期がタイトであったため、直接案件対応にかかわる顧客・内装設計担当及び生産部署との縦の連携が重視され、電装設計部署内での設計者同士の横の連携が疎かになりがちで、情報共有がうまくいっていなかった。
それが原因でミスを誘発、又は自分以外の担当者が過去に発生させたミスを再発させてしまうなどの事例が少なからず存在した。

・ＣＡＤ導入以前の古い手書き図面や、逆にリリース前の最新モデルのラフスケッチのような検討図などが多数存在していたため、図面間の比較や干渉の確認がしにくく、ＣＡＤ上で検討するために都度データ化する必要があり、時間がかかる

・生産工程までいかないと設計の成否が確認できない

＜CASE.3における改善案＞

問題点を纏めると以下の通りである。

・設計者間での知の共有が不十分であった。

・使用しているツール（2D-CAD）が、現状の問題改善に対応しきれていない。

この問題の改善案については以下の２つになる。

（１） 顧客固有ルール、特殊な対応を行った案件のデータベース化

（２） ３Ｄ－ＣＡＤの導入、非データ図面のデータ化

（１）について、ダンプの場合と同様に艤装オプションの項目は多岐にわたり、完全な標準化は出来ない。よって、ミスのトラブルシューティングを主とし、過去の案件で実際に起こったミスとその対応を閲覧出来るようにすることを目的としたデータベースをExcelで構築する提案をした。

（２）について、設計工程での問題点洗い出しの正確性を改善するため、３Ｄ－ＣＡＤを用いて実際の機器配置モデルを確認し設計段階でミスをつぶすこと、また、紙図面のＣＡＤデータ化を随時行うことでその場その場での対応を無くし、納期計画への影響を抑える提案をした。

＜CASE.3における実践＞

（１）について、データベースの構築において、実際に起こったミスの事例や設計者が行った特殊な対応についてデータを収集した。結果、情報が伝達されておらず、設計者間で共有されていないことが多いということが改めて分かった。このことから、データベース化することと並行し定期的な設計チーム内会議の開催を行い、各担当の案件の進捗や問題点など情報共有を行うことが決まった。データベース化や図面のデータ化にも本格的に取り組み始め、2～3か月でミスの発生率が2～3割程度減少し、ミスが発生した場合でも過去の類似事案を検索しやすくなったことから対応精度・対応速度も向上した。

（２）について、３Ｄ－ＣＡＤの導入も試験的に行われ、車体のモデルチェンジ時の設計変更点の洗い出しや部品干渉の確認などに利用され、設計段階においてより具体的な変更案の提案が可能となった。

＜CASE.3における今後の課題＞

入した改善案は一定の成果を上げることが出来た。しかしこれはコロナ禍による生産台数減少時だから出来た対応であるとも考えられる。コロナ禍が収束し以前の様な生産計画に戻った時、同様な対応を継続して行えるかが課題と考える。

自身は大学院入学に伴い現場から離れたため結果はわからないが、恐らく設計部署のみの取り組みでは難しく、営業部署・生産部署など絡めたアジャイル方式の開発方法論の構築が必要になってくると思われる。

情報システムの開発方法論

システム開発には代表的な開発モデルとして「ウォーターフォール方式」と「アジャイル方式」という二つのモデルがある。

・ウォーターフォール方式

システム開発における工程を初めから終わりまで順番に行う方法で、システム開発における最も基本的なモデルである。各フェーズの終わりに検証を行うことで、次のフェーズにできるだけバグを持ち込まないようにする流れとなる。

・アジャイル方式

システムの大まかな仕様を決定し、システム全体を1週間程度で開発できる規模の機能に分割する。分割した機能ごとに、要件定義、設計、開発、テストをし、完成したら機能の単位でリリースする。この一連のサイクルをイテレーションと言い、イテレーションを繰り返してシステムを完成させる方式である。

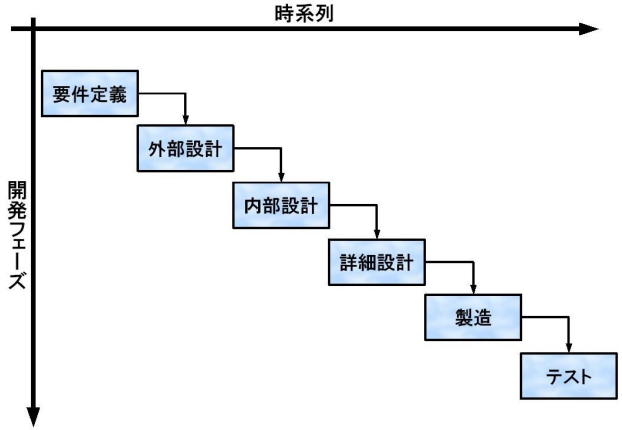


図9　ウォーターフォール開発

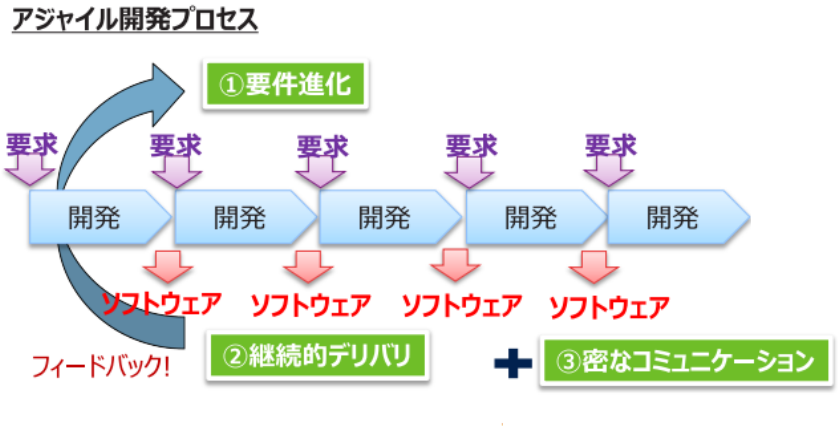


図10　アジャイル開発

・DevOps

アジャイル方式を実践する場合、並行して導入されることもある概念。開発チームと運用チームが、ビジネスを成功させるという一つの目的のために協力し、サービス価値の仮説を素早く検証しながらユーザーニーズに近づく手法。

取り組み自体に関しては、アジャイル方式と同質のものである。

・フロントローディング

プロジェクト早期の段階で施主や設計者の意向を取り入れて合意形成を行い、製品開発の初期工程にリソースを投入することでコスト、品質を作りこみ、前倒しで作業を進めることで早期に施工図や施工計画を完成させる活動を表す。

フロントローディングを行うときには、ツールを効率的に活用してシミュレーションをコンピュータ上で行う、ビュワーなどを使用することにより後工程の部門との連携を容易にすることなどが同時に必要となってくる。

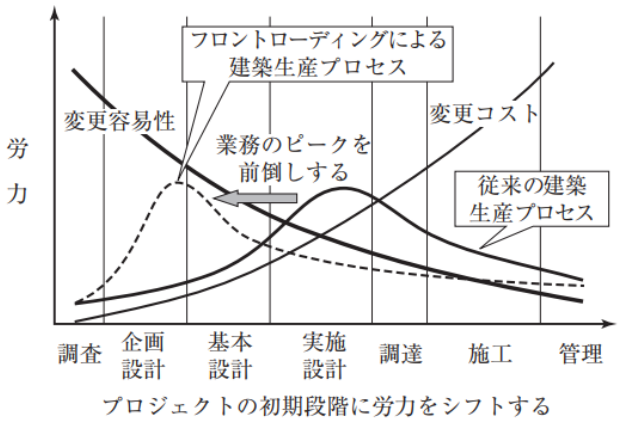
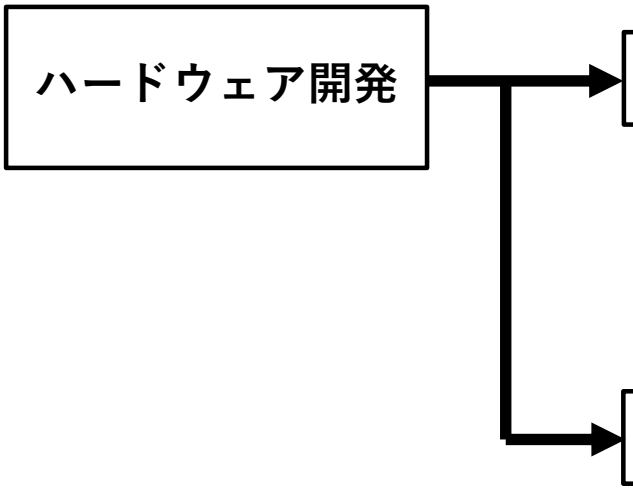


図11　フロントローディング

アジャイル方法論に基づくCASE.1～3の改善案



＜受注設計におけるアジャイル方式を考えた場合＞

・ハードウェア設計では一つのモノとして製造・納品が行われるため、顧客への分割納品が出来ない。
・機能チェックを行う際もデータ上での検討はあくまで理論値で、製品の質感や操作感などは実際に製造したものに対してしか行えない。
・製造工程において多くの場合専用の大型製造機械が必要となり、生産作業をライン工程で行うことになる。途中で製品をピックアップして再検討をすることはラインを止めることになるため、時間的なリソースを大きく消費することになる。

上記より、受注生産設計においては「機能の単位でリリース・検討」は有効ではなく、
製造工程前の図面における「設計検討段階でのイテレーション」が有効である。

CASE.1～3においては、受注設計は設計期間がタイトな場合がほとんどで、顧客を交えた密な打ち合わせを行うことは難しい環境にあった。しかし、これを実現することで設計段階でのミスを減らして製造工程に影響が出ない様にすることは、コスト削減の観点から見て効果的であると考えられる。

＜開発設計におけるアジャイル方式を考えた場合＞

・開発設計においては試作機での検討となるので、通常生産ラインへの影響はさほど大きくない

・モデルチェンジ開発の場合、明らかに前バージョンと異なる部分というのは仕様の時点で分かっており、前バージョンを踏襲できる部分は少なくない

・試作品のため、要所のみの分割製造・実物での検証は現実的に可能

上記より、開発生産設計においては「機能の単位でリリース・検討」は有効であり、
製造工程前の図面における「設計検討段階でのイテレーション」も受注設計と同様に有効である。

＜フロントローディングの導入＞

・ハードウェア開発において、不具合修正による製造ラインストップや再制作のための部品の手配・作成などのリソース損失がロスの大部分を占めるため、設計段階で十分に検討を行うことは有効な対応策である（実感としてCASE.1及び3において効果が認められる）

「初期工程段階でリソースを投入し品質を向上させる」というフロントローディングの考え方はハードウェア設計において有効であると考えられる。

＜アジャイル方式導入における課題＞

・アジャイル方式導入は生産コスト削減という面から改善案として適当であると考える。
　しかし、時間的・組織的なコストの増加については考慮が必要となる。

・フェイス・トゥ・フェイスの情報伝達を実現するための改善について、自身の体験としても
　問題点に挙げたが、組織的な刷新が必要となり実現が困難であったため、改善はされなかった。

・当時はビデオ会議が一般的でなかったため、組織的な刷新が必要となり実現が困難であったが、
　現在においては現実的に改善可能であるため、遠隔会議システム導入を積極的に進めるべきと考える。

提案

前項までを踏まえ、ハードウェア設計における受注設計のアジャイル方式のプロセスモデルを図12に示す。
（開発設計については前述の図10のモデルが適応可能と考える）

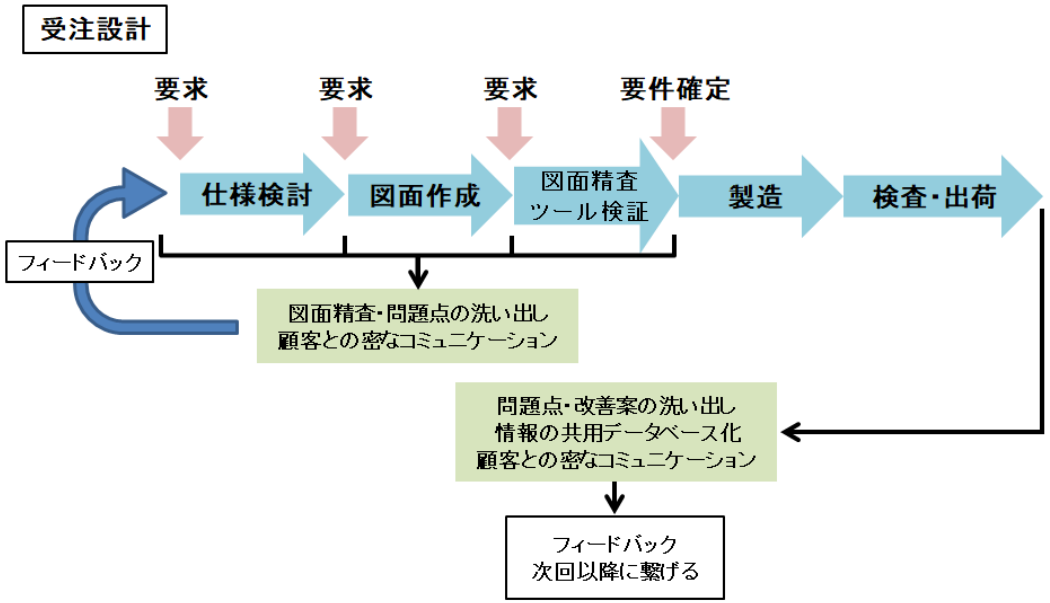


図12　ハードウェア設計（受注設計）における
アジャイル方式の導入モデル

結論

自身の経験に基づきハードウェア設計プロジェクトの問題点を洗い出し、
アジャイル方式を適応出来るところと出来ないところを考察し、モデル提案した。

ハードウェア開発は、ソフトウェア設計と違い品質チェックを行う際に製造工程が関わってくるため、
完成品に対する短期間の繰り返しチェックよりも、図面での設計検討段階におけるイテレーションの繰り返し・フロントローディングの導入が改善案としてより効果的と考える。

設計者は限られた時間の中で、顧客・部署・設計者間での情報共有、効率的なツール運用技術の取得等をどれだけ意識的に改善に取り組んでいけるか、
企業は関係者間で情報を密にするスムーズな情報伝達網の構築や、アジャイル方式導入におけるリソースの確保を、現状維持から改善に向けてどう対応していくのかが重要課題である。