

人工物作成コミュニケーションの比較についての考察

～住宅建築とソフトウェア開発の場合～

山本 佳和[†] 山本 修一郎^{††}

[†]株式会社デンソークリエイト プロジェクトセンター システム1室
名古屋市中区栄3-1-1 広小路第一生命ビル

^{††}名古屋大学 情報連携統括本部 情報戦略室
愛知県名古屋市千種区不老町

Consideration about comparison of communication on development of artificial material

Yoshikazu YAMAMOTO[†] and Shuichiro YAMAMOTO^{††}

[†]SOFTWARE DEVELOPMENT DEPT. Denso Create Inc.
Sakae, Naka-Ku, Nagoya Aichi Japan

^{††}Strategy Office, Information and Communication Headquarters, Nagoya University
Furo-cho, Chikusa-ku, Nagoya Aichi Japan

概要

ソフトウェア開発では、様々な面で失敗例が多く報告されている。これら失敗はソフトウェア開発特有のものなのか、また、それはどんな特殊性によって引き起こされているのか。また、とりわけ失敗の原因が開発関係者間のコミュニケーションであるとされる場合も少なくない。本稿では、同じ人工物を開発している住宅建築の場合と、ソフトウェア開発を比較することにより、ソフトウェア開発が潜在的に抱えている問題点と開発コミュニケーションの関係を考察する。

Abstract

Many failure cases have been reported in various aspects in software development. Were these caused by some peculiarity of software development? If it was so, what kind of characteristics were these caused by? It is often said that the reasons for the failures were communications among development stakeholders. In this paper, we compare communications between software development and residential construction. The reason to select residential constructions is that it is the artificial product development similar to software development. Then we also consider the relationship between the potential problems of software development and the development communications.

1 はじめに

ソフトウェア開発は、様々な場面での不確実性から、問題が発生し、結果として、本来の目的と提供機能の乖離、開発予算、開発期間の超過など、多数の失敗例が後を絶たない。これら失敗例が何故起きたのか、起こさないためにどうすれば良いのかという議論は、それぞれの個別の問題を取り上げただけでも簡単に解決できるものではない。とりわけソフトウェア開発での失敗の原因は、開発関係者間のコミュニケーションが原因とされることも少なくない。これらコミュニケーションに起因する問題は、ソフトウェアという人工

物特有の問題なのだろうか。そうであれば、他の人工物作成の場面では、これら問題をどのように回避しているのだろうか。本稿では、同じ人工物の開発を行う住宅建築の場合と、ソフトウェア開発とを比較し、その相違点を観察することにより、ソフトウェア開発が抱える問題点を考察する。住宅建築の場合を取り上げた理由は、ソフトウェア開発に関係するメンバが、注文者の立場で関係した経験が高そうな分野であることによる。

2 住宅建築とソフトウェア開発の比較方法

住宅建築とソフトウェア開発の比較に際して、より客観的な比較となるように、開発中の関係者間のコミュニケーションについての比較だけではなく、複数の観点を設定する。今回の比較に際しては、以下の観点から比較する。

(1) 開発工程

開発物の注文者が、その開発物の必要性を感じてから、その開発を依頼し、開発が完了するまでの流れや、開発工程を比較する。

(2) 開発文書

開発の途中で作成される文書の種類と、その利用方法

について比較する。

(3) 構成要素

開発物を構成している主な要素と、その役割、標準化度合いを比較する。

(4) コミュニケーション

開発途中での注文者と開発者、開発者同士のコミュニケーションの内容を比較する。

3 工程の比較

注文者が、注文対象の必要性を感じてから、注文し、開発を完了するまでの流れと工程について、表 1 に示す。

表 1 開発工程と工程内の活動内容の比較

| 開発工程 | 住宅建築 | ソフトウェア開発 |
|-------|---|--|
| 引き合い | <ul style="list-style-type: none"> ■ 住みたい家のイメージづくり <ul style="list-style-type: none"> ● 間取り、設備などと生活状況を想像 | <ul style="list-style-type: none"> ■ 欲しい機能の要望 <ul style="list-style-type: none"> ● 業務上の課題、問題をなんとかしたいという思い |
| 土台決め | <ul style="list-style-type: none"> ■ 土地探し <ul style="list-style-type: none"> ● 広さ、方角、ライフライン整備状況の検討 ● 周辺環境と生活イメージとの合致状況の確認 | <ul style="list-style-type: none"> ■ 運用環境の特定 <ul style="list-style-type: none"> ● どのシステムで動くものかを決定 ● どのように使いたいかの想定 |
| 開発者選定 | <ul style="list-style-type: none"> ■ 建築業者の選定 <ul style="list-style-type: none"> ● ハウジングセンター巡りをしたり、新聞、雑誌、Web などからイメージに合うメーカーを選別 | <ul style="list-style-type: none"> ■ 開発会社選定 <ul style="list-style-type: none"> ● 開発実績のある会社や、関係会社への打診 |
| 要求の確認 | <ul style="list-style-type: none"> ■ 要求の伝達 <ul style="list-style-type: none"> ● どんな家に住みたいのかという目的の伝達 ● 予算、入居希望時期、将来も想定した家族構成などの伝達 ● 多くは、開発側からのヒアリング ● 場合によっては工法の指定 | <ul style="list-style-type: none"> ■ 要求の伝達 <ul style="list-style-type: none"> ● 困っていること、やりたいことの伝達 ● 予算、納期の伝達 ● 自身の経験や、一般に使われているソフトウェアを例にとつての機能の提案 ● 動作環境（対象システム、OSなど）の指定 |
| 外部設計 | <ul style="list-style-type: none"> ■ 注文者の要望、イメージの確認 <ul style="list-style-type: none"> ● 平面図などにより、部屋割り、広さ、生活動線などの確認 ● 水道、電気、ガスの位置確認 | <ul style="list-style-type: none"> ■ 実現機能の確認 <ul style="list-style-type: none"> ● 注文者の要求に対して、実現する機能を定義 |
| 内部設計 | <ul style="list-style-type: none"> ■ 注文者の要望、イメージの確認 <ul style="list-style-type: none"> ● 立面図などにより、ドア、窓などの種別、位置、サイズ確認 ● 建築部材などの選定 | <ul style="list-style-type: none"> ■ 実現方法の決定 <ul style="list-style-type: none"> ● 効率的な内部構造や、主要 I F の定義 ● データ構造の決定 |
| 実装 | <ul style="list-style-type: none"> ■ 工務店などによる実建築 <ul style="list-style-type: none"> ● 図面どおりの指定部材による指定寸法での組立て ● 工法/技法が業界として確立しており、一定の知識があれば、ほぼ同じ品質レベルで対応可能 | <ul style="list-style-type: none"> ■ プログラマによるコーディング <ul style="list-style-type: none"> ● 場合によっては設計内容から、ツールによって、ソースコードを半自動生成" ● 詳細アルゴリズム、例外処理などの作成 ● 一定の言語知識等があっても、品質レベルは人によってばらつく |
| 検査 | <ul style="list-style-type: none"> ■ 第三者機関による検査 <ul style="list-style-type: none"> ● 法的に定められた基準に従い、第三者が検査を実施 | <ul style="list-style-type: none"> ■ 開発者が作成したテスト仕様に基づく検査 <ul style="list-style-type: none"> ● 明確な基準は無く、開発者の判断による品質レベルの検査を実施 |

住宅建築もソフトウェア開発もおおまかな工程分類と、その順序は同じである。しかし、住宅建築の場合、各開発工程での作業内容、確認観点や、基準が整備さ

れており、不明確なものは注文者の要求などに局所化されている様子がわかる。このため、開発工程では、注文者の要求を明確にするという作業が中心となっ

ている。明確にした要求を起点として、確立されている原理・原則、基準、技法をあてはめて開発に向かう流れである。

一方で、ソフトウェア開発では、ソフトウェアで実現する機能を定義することに注力している。現実的に開発期間や、予算制約が厳しい中で、ソフトウェア開発には、まだまだ開発作業に不明確な要素が多い。この

ため、開発者自身の経験や、一般に使われているソフトウェアを例にとりて明確化しやすい機能や、制約事項を決め、開発に向かう流れとなっている。

4 文書の比較

開発の各工程で作成する文書と、その利用方法の比較を表 2 に示す。

表 2 開発時に作成する文書と、その利用方法の比較

| 開発工程 | 住宅建築 | ソフトウェア開発 |
|------|--|---|
| 要求分析 | <ul style="list-style-type: none"> ■ カタログ ● イメージづくり | |
| | <ul style="list-style-type: none"> ■ 打ち合わせ議事録 ● 要望など確認/合意内容の記録 | <ul style="list-style-type: none"> ■ 打ち合わせ議事録 ● 機能要望、制約など確認/合意内容の記録 |
| 外部設計 | <ul style="list-style-type: none"> ■ 平面図（フリーハンド） ● 概要の確認と共有 ※ 現地で手直し可 | <ul style="list-style-type: none"> ■ 外部仕様書 ● 実装機能の一覧を定義 ● 特にフォーマットは決まっていない ● ユーザ（利用者）は「機能一覧」程度しかみないもしくは、まったく見ない |
| | <ul style="list-style-type: none"> ■ 平面図 ● ドアの開閉干渉、間取りの確認 ● 生活動線の確認 ● 窓の位置、大きさの確認 | <ul style="list-style-type: none"> ■ ユースケース図 ● 開発システムの範囲を明記し、アクタに対して、システムの振る舞いを記述する ● 注文者にはイメージがつきにくい面もあり、開発者がシステムの範囲を明確に整理するために利用する人が多い |
| | <ul style="list-style-type: none"> ■ 立面・断面図 ● 外観の確認、立地状況の確認 | |
| | <ul style="list-style-type: none"> ■ 配置図 ● 土地の中での建物の位置、外からのアプローチ（動線）の確認 | |
| | <ul style="list-style-type: none"> ■ 外部仕上げ表 ● 外装品の材料/部材一覧。材質、断熱材の利用など、利用材料、工法などの確認 ● 業界の共通仕様なども存在 | |
| | <ul style="list-style-type: none"> ■ 内部仕上げ表 ● 内装品の仕様確認 | |
| 内部設計 | <ul style="list-style-type: none"> ■ 平面詳細図 ● 具体的にどこに、何が、どのように作られるのかを確認 | <ul style="list-style-type: none"> ■ クラス図 ● 主要クラスの関係のみのことが多い |
| | <ul style="list-style-type: none"> ■ 電気設備図 ● 電気、コンセントの位置を確認 | <ul style="list-style-type: none"> ■ シーケンス図 ● 主要シーケンスのみのことが多い |
| | <ul style="list-style-type: none"> ■ 給排水衛生設備図 ● 水周りや、元栓位置などの確認 | <ul style="list-style-type: none"> ■ 状態遷移図 ● 主要な状態遷移のみのことが多い |
| 実装 | <ul style="list-style-type: none"> ※ 実現人工物は、「文書」ではない | <ul style="list-style-type: none"> ■ プログラム ● 実現人工物そのもの |
| 検査 | <ul style="list-style-type: none"> ■ 壁量計算書 ● 設計配置した耐力壁の量が、建築基準を満たしているかどうかを確認する ● 主に、第三者検査機関の確認用。 ■ 構造計算書 ● 第三者検証機関が、検査に際して、必要と指定した図書 ※ 様々な基準が存在し、その基準にしたがっていることを表現できる図書などがある | <ul style="list-style-type: none"> ■ テスト仕様書 ● テストケースが羅列されている ● 特にフォーマットは決まっていない ■ テスト報告書 ● テストした結果が記載されている ● 特にフォーマットは決まっていない |

この比較から、まず、明確にわかることは、開発工程内で作成する文書種類の差である。また、住宅建築で作成する図面類はすべて明確な記法が定められており、その用途も明確に定義されている。また、これら図面を作成する担当者は、一定の知識教育を受けて、建築士等の資格を得た者であるため、各図面の記述内容は、それぞれの基準に明確に準拠できているといえる。

しかし、ソフトウェア開発では、文書種類はある程度の定義があるものの、その記述内容、形式は各開発組織によって異なっている。UML などの記法も存在するが、開発にあたり、これらの知識、記法を開発者が有していることは義務付けられていない。このため、UML 等の標準記法を用いても、その記述の正確さは開発者により、ばらついてしまう。また、UML の各図も、用途は定義されているものの、明確に開発者に意識されているわけではないため、作成しても組織によっては有効性が低く、使われない場合もあり、作成されなくなっていく。

また、各文書の内容を比較すると、住宅建築の場合は圧倒的に図面であり、文章量は少ない。ソフトウェア開発文書は文章量が多く、書き手の意図を詳細伝えるということが出来る。しかし、明確な記法が定まっていないため、書き手の文章力に依存する。

住宅建築の文書で、もうひとつ特徴的なのは、その図面が

幅広い関係者に理解可能ということである。注文者、建築業者など、多くの関係者が、それぞれの視点で出来上がる成果物のイメージを共有できる。

しかし、ソフトウェア開発では、記法が定義された図は、主に開発者向けのものとなっており、注文者が理解できないケースも多い。記述された文章も、その文章量の多さから、大抵のケースでは注文者に読まれないこともあり、出来上がる成果物のイメージが概要レベルでしか共有できない。

また、住宅建築では、「何が安全か?」「何が快適か?」などの尺度が明確化されていることもあり、それにともない、検査基準も明確になっているため、検査図書も充実している。このため、第三者による客観的な検査も可能となっている。

5 構成要素の差異

開発物を構成する主要素と、その役割や標準化度合いの比較を表 3 に示す。

表 3 開発物を構成する主要素の比較

| 住宅建築 | ソフトウェア開発 |
|--|--|
| <ul style="list-style-type: none"> ■ 土地 ● スペースは確定、地盤の強さも測定可能 | <ul style="list-style-type: none"> ■ メモリ ● スペースは確定 |
| <ul style="list-style-type: none"> ■ 基礎 ● 地盤が弱い際に、いかに上モノを支えるかを考え、どう地盤を強くするかがほぼ決まっている。 ● 上モノにあわせて、必要な強度設計が可能 | <ul style="list-style-type: none"> ■ OS ● 上モノに対して共通の機能を提供 ● 上モノはハードウェアを意識しなくてすむ ● 上モノに対して最適かどうか評価が困難 |
| <ul style="list-style-type: none"> ■ 骨格 ● 工法と、その長所/短所が明確になっている | <ul style="list-style-type: none"> ■ フレームワーク ● 長く使われ続けているものが少なく、一定期間で世代交代していく |
| <ul style="list-style-type: none"> ■ 外装・内装 ● 基本コンポーネントは同じで、用途、好みに合わせて装飾可能。 | <ul style="list-style-type: none"> ■ UI・アルゴリズム ● 基本となるアルゴリズムはいくつかあるが、用途、制約条件によって、個別に開発/改良している |
| <ul style="list-style-type: none"> ■ 設備 ● JIS などの規格でサイズなどが決まったコンポーネント | <ul style="list-style-type: none"> ■ 部品・市販コンポーネン ● 様々なものが提唱されているが、規格はないため、取替えは簡単ではない場合もある |
| <ul style="list-style-type: none"> ■ 什器 ● 用途に合わせて各メーカーが発売しているものから選ぶ。インターフェースは JIS などの規格で定義できているため、規格に合えば組みつけ可能 | <ul style="list-style-type: none"> ■ 外部アプリ ● 様々なアプリで共通 I F を提供しているものはまれであるため、欲しい機能を提供するアプリ独自の実装が必要 |

構成要素についても、住宅建築、ソフトウェア開発ともに、その分類自体に大きな差はない。ただし、大きく異なるのは標準（規格）／基準の存在にある。住宅建築では、明確

な標準（規格）が存在する。このため、住宅建築においては、これら標準（規格）に従って成果物を作成していくことで、様々な装飾、設備、什器などを選択できる。しかし、

ソフトウェア開発の場合は、これら標準（規格）が少ないため、基本的には機能によってコンポーネントを選定し、母体ソフトウェアの開発側がコンポーネントのIFにしたがって、成果物を開発する流れが一般的である。このため、コンポーネントの取替えが困難となるケースもある。

6 コミュニケーションの比較

開発途中での注文者と開発者、開発者同士のコミュニケーション状況の比較を表4に示す。

表4 開発中のコミュニケーション状況の比較

| 開発工程 | 住宅建築 | ソフトウェア開発 |
|------------------------|--|---|
| 引き合い ～開発者選定 | <ul style="list-style-type: none"> ■ 営業と注文者 ● モデルルームなどで、最高ランクの住宅を見せて説明し、注文者のイメージを探る ● 営業訪問時は、カタログの紹介をしながら、雑談を含めた会話が主体で、注文者の趣味、嗜好、事情などをしっかりメモしている | <ul style="list-style-type: none"> ■ 営業もしくは、技術リーダーと注文者 ● 過去の開発実績で持っている技術、品質管理体制などを説明し、注文者に安心感を与える ● 背景などを聞きながら、その機能が必要となった経緯をメモしている ● 納期、予算などの制約事項を気にするケースが多い ● 注文者が開発システムの利用者で無いケースが多い |
| 設計 (要求の確認 ～内部設計) | <ul style="list-style-type: none"> ■ 設計担当者と注文者 ● 家のコンセプト・目的を決めるため、注文者の家に対する要望、イメージなどをひき出す ● 引き出した要望を図面の形で表し、イメージを具体化していくことにより、より詳細な要望を探っていく。また、注文者とは具体的な図面を通して作りたいものの認識をあわせていく ● この期間が長く、定期的に図面を介しての打ち合わせがある。また、打ち合わせごとに図面が詳細化、立体化されていき、時にはショールームで実物を見ながらの打ち合わせを行う | <ul style="list-style-type: none"> ■ 技術リーダーもしくは、設計担当と注文者 ● 欲しい機能の詳細をヒアリングし、どんな機能が必要かをひき出す ● 引き出した要求を元に、機能の一覧を定義した外部設計書を作成する。注文者は長い文章を読まないため、機能の概要程度に目を通して合意するケースが多い ● 定期的な打ち合わせを実施するケースでは、主に作業進捗報告や、開発期間、開発費の見込み状況変化などについて開発側から報告する |
| 実装 | <ul style="list-style-type: none"> ■ 営業と注文者 ● 工事の進捗状況の報告が主体 ■ 設計者と実装者 ● 開発に必要な情報は、基本的にすべて図面に記載されているため、図面を介した会話となる。主に、現場の安全性や、仕上げ作業の丁寧さについての会話が主体 | <ul style="list-style-type: none"> ■ 技術リーダーと注文者 ● 開発の進捗状況の報告が主体 ■ 設計者と実装者 ● 設計書にすべて記載されていないため、設計意図の確認が主体 |

「3 工程の比較」でも触れたが、住宅建築の場合、注文者が持つ、成果物に対するイメージを引き出すためのコミュニケーションに重点を置いている点がわかる。注文者にも理解できる図面を通して、イメージの確認を繰り返しているところに特徴がある。また、イメージの引き出しと、図面の具体化のプロセスが周期的に繰り返されることが、注文者とのイメージ共有を確実なものとし、同時に設計図書が完成するという流れになっている。

これに対し、ソフトウェア開発では、成果物に対する注文者とのイメージの共有は、実現機能の概要レベルでの定義となるケースが多い。ソフトウェア開発の中でもプロトタイプ開発など、成果物をユーザに見せながらイメージを共

有していく方法がとられることはある。しかし、住宅建築のように、イメージ共有のゴールと、設計完了のゴールが一致していないために、開発期間や、予算制約の面から、多くの場合、必ずしも狙った効果が得られていない。

また、住宅建築では、設計工程の終了時点で、図面上では、対象の建築物は完成している。このため、後工程の実装者は、図面どおりに作っていけば、上流工程で指示した建築物を完成することができる。実装工程に入った後の設計者と実装者、実装者同士の会話は、建築現場の安全確保や、図面上に出てこない細かな仕上げ作業に関するものが主体となる。

一方、ソフトウェア開発では、主要な責務分担や、シーケ

ンスの定義しか設計書に記載されない。このため、後工程にはいっても、実装者と設計者の間では設計意図の確認などの会話が多数。また、実装者同士も実現方法の相談などの話題が中心となる。

7 関連研究

Berryも住宅建築とソフトウェア開発の類似性を指摘している。[1]とりわけ、要件定義時の注文者と開発者の関係や、その定義手順に類似性を認めている。しかし、その中で、両者の大きな相違点として、要求の変化への対応を挙げている。住宅建築では、要求の変化に応じて、開発予算や、開発期間の見直しが行われる。また、これらへの影響を抑えたい場合は、注文者も要求の変更に対して慎重になると述べている。これに対し、ソフトウェア開発の場合は、要求の変更に対して、当初予算、開発期間のままで開発することが当然であるかのように扱われていると述べている。また、別の論文では、建築業界での要求定義を例に、要求定義により多くの時間を費やすことが、後の開発を円滑にすすめることにつながるとしながらも、実際のソフトウェア開発では、そこまでの期間を費やすことが不可能であるとも述べている。[2]

8 まとめと今後の課題

本稿では住宅建築とソフトウェア開発を開発工程、活動内容、開発文書とその利用方法、開発物の構成要素、開発コミュニケーションの状況に基づいて比較した。その結果、注文者の不確実な要求を工程を進めるにしたがって、形にしていくという活動は同じであった。しかし、両者の相違点という部分では、Berryが述べたような、要求変化に対する認識や、要求定義に費やすことのできる期間の問題だけではないと、著者らは考える。住宅建築にかかる期間は、通常のハウスメーカーへの委託の場合、半年～1年程度である。これは、ソフトウェアシステムの開発期間と比べた場合、決して長い開発期間とはいえない。筆者らは、この開発期間での、両者の活動の仕方に注目したい。今回の比較結果から、住宅建築の場合は、開発途中で曖昧になりがちな部分をしっかりと定義し、明文化できていることがわかる。これにより、関係者の認識を、より簡単に合わせる事が出来ている。その結果、注文者の要求など、本質的に不明確な部分に焦点をあてて、認識を共有するという高

品質なコミュニケーションが保証できている。一方、ソフトウェア開発では、一般的に定義しておくべき規格であったり、開発図書といった部分の整備が遅れている。この結果、本質的な部分に時間をかけられず、関係者との認識の共有が曖昧になるというコミュニケーション品質の低下を招いている。したがって、ソフトウェア開発においても、住宅建築の場合と同様に、本質的に不明確な部分と明確に出来る部分を、しっかりと切り分ける手段を環境として整備していくことで、コミュニケーションを高品質化できる可能性がある。

本稿での考察から、ソフトウェア開発コミュニケーションの高品質化に向けて、住宅建築に比べて決定的な違いがあらわれた開発文書を整備することが必要であるといえる。開発文書を多くの関係者を対象としたコミュニケーションの道具とするためには、以下の観点で文書種類、体系を見直す必要がある。

- (1) コミュニケーション対象者の特定
- (2) コミュニケーションに必要な情報の特定
- (3) コミュニケーションを通して合意形成をする上で必要な、知識レベルを合わせた表現方法の特定

ソフトウェア開発活動の中での開発文書の位置づけや、文書としての品質定義および、測定手法を見直す組織として、システム開発文書品質研究会 (ASDoQ)^[3]がある。本研究会のロードマップ部会では、開発文書の情報価値としての整理、文書としての価値定義や品質特性と、その定量的な測定法の定義など、開発文書体系を見直すためのロードマップを策定している。この部会の活動を通して、今回の考察から得られた課題の解決に取り組んでいきたい。

参考文献

- [1] Berry, D.M., “Software and House Requirements Engineering: Lessons Learned in Combating Requirements Creep”, *Requirements Engineering Journal* 3(3&4), pp. 242–244 (1998).
- [2] Daniel M. Berry, The Inevitable Pain of Software Development: hy There Is No Silver Bullet, 10 August 2003
- [3] システム開発文書品質研究会ホームページ, <http://asdoq.jp/>