

変更の影響範囲を特定するための「標準調査プロセス」の提案

Unified investigating process to Detect the range Affected

by specification changes

主査 : 飯泉 紀子 (株式会社日立ハイテクノロジーズ)
副主査 : 足立 久美 (株式会社デンソー)
アドバイザー : 清水 吉男 (株式会社システムクリエイツ)
リーダー : 宇田 泰子 (アンリツエンジニアリング株式会社)
研究員 : 夢田 一成 (アンリツエンジニアリング株式会社)
川井 めぐみ (サントリーシステムテクノロジー株式会社)
伊藤 友一 (T I S株式会社)

研究概要

派生開発では、頻繁に発生する変更要求を、限られた時間内で対処することが求められる。時間内に効率よく漏れなく変更による影響範囲を特定することが、品質・コスト・納期(QCD)の目標達成、ひいてはビジネスの成否をも左右する。現場でどのように変更の影響範囲を特定しているのかをヒアリングしたところ、調査方法や調査結果の残し方が様々であることがわかった。これでは調査の質の安定は望めない。また、現場で行なわれている調査の仕方と派生開発に特化したアプローチである XDDP (eXtreme Derivative Development Process) のプロセスを比較した。その結果、ソースコードを処理の順番に沿って探索していることが違いとしてわかった。この探索法では、調査範囲が狭くなり、本来目的別に分けて行なうべき調査を区別なく行なっているため、影響範囲の特定漏れを引き起こしやすくなるという課題が見えてきた。この課題を解決するために、調査を事前調査、変更箇所調査、影響調査という3つのステージに分けた「標準調査プロセス」を作成した。これを現場で有効に活用してもらうために「調査プロセスガイドライン」を作成した。これらを使って過去の事例でシミュレーションしたところ、変更の影響範囲の特定漏れを減少させることができた。

Abstract

In derivational development, developers need to meet frequent requirements of changes on time. Detecting the range affected by specification changes efficiently and without omission is a must to achieve the goals of quality, cost and delivery (QCD) and therefore to succeed in our business. Now we took a survey from our developers and found out that each developer finds the parts in range affected by specification changes need to be changed and keeps records of those in various ways. In this way, we cannot expect stable high quality. We compared XDDP (eXtreme Derivative Development Process), which is supposed to be an effective approach in derivational development with development process that has been done in the field. And we found that there is a difference in the search of the source code. In addition, we found that there is a problem with this search method. In order to solve these problems, we defined a "standard survey process" for detecting the range of influence changes and summarized the use of the research process as a "Survey Process Guide". We tried this procedure in a project whose tests have failed, and we succeeded in reducing the number of errors in finding the range affected by specification changes.

1. はじめに

既存システムへの機能追加や部分的な変更を行なう派生開発においては、様々な内容の

第6分科会（Aグループ）

変更要求が頻繁に発生し、迅速な対処が求められる。しかし実際には、変更による影響範囲を特定できないことに起因する不具合が後を絶たず、不具合修正のための手戻りにより予定工数を超過するケースが発生している。プロジェクトのQCDを確保しながら、様々な内容の変更要求に対処するためには、効率よく漏れなく、変更による影響範囲を特定できるようにしなければならない。

派生開発の現場で発生している不具合事例を分析したところ、現場で行なわれている調査方法が、担当者によって差があることがわかった。また、調査結果としてメモ書きしか残せていないため、レビューにおいて調査の抜け漏れに気が付きにくいこともわかった。我々は、解決策として、調査方法および調査結果の人によるバラツキ（属人性）を排除し、調査プロセスの水準を高めることを目的として、調査方法の統一および調査結果のテンプレート化（レビューインプットの明確化）を行なった。これを「標準調査プロセス」として定義した。さらに「標準調査プロセス」を現場で有効に使用してもらうため、「調査プロセスガイドライン」としてまとめた。検証の結果、「標準調査プロセス」を適用することで、適切な調査のステージで適切な調査を行なえた。また、レビューのインプットになる調査結果を残すことができ、調査の抜け漏れを防ぐことができた。

以降、第2章では調査における現状の問題点を分析し課題を整理する。第3章では、解決策である「標準調査プロセス」と「調査プロセスガイドライン」を提案する。第4章では、「標準調査プロセス」と「調査プロセスガイドライン」の適用結果について述べる。第5章では、提案の評価を行なう。

2. 現状分析

2.1 不具合事例の分析

我々は研究員の現場でどのような不具合が発生しているのか、115件の不具合事例を集め、以下に示すように4つのカテゴリに区分し、分析した。カテゴリ別の発生件数とカテゴリ別の手戻り工数を図1に示す。

(A) 変更要求の問題

変更要求を正しく捉えなかったことに起因する不具合

(B) 変更内容の問題

変更仕様を正しく設計できなかったことに起因する不具合

(C) 調査方法の問題

変更の影響範囲の特定、変更による影響を正しく洗い出すことができなかったことに起因する不具合

(D) 実装の問題

正しく実装できていなかったことに起因する不具合

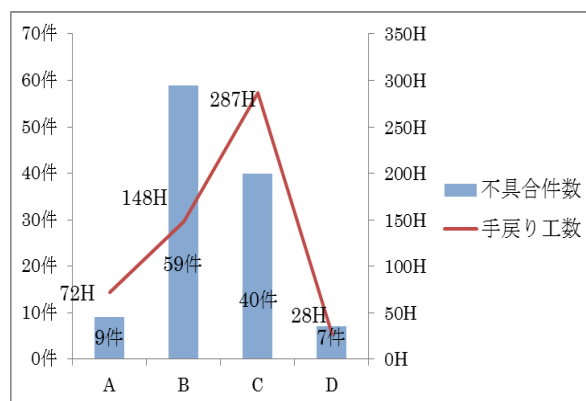


図1 不具合件数と手戻り工数

この分析結果から、不具合件数の第1位が「変更内容の問題」の59件、第2位が「調査方法の問題」の40件、手戻り工数の第1位が「調査方法の問題」の287時間、第2位が「変更内容の問題」の148時間となることがわかった。「変更内容の問題」の発生原因の多くは「ソースコード上で変更箇所を見つけ次第に変更し、十分な変更仕様の検討が行なわれていない」ことが原因であることから、既存の技術であるXDDP[1][2][3]を適用することで解決可能であることがわかっている。そこで我々は、不具合件数が「変更内容の問題」に対して若干少ないが、手戻り工数が第1位の「調査方法の問題」を解決することで、プロジェクトの品質・コスト・納期（QCD）に与える影響の低減を図ることにした。

第6分科会 (Aグループ)

2.2 変更の影響範囲の調査プロセスの実態

派生開発において変更の影響範囲をどのように調査しているのか、その実態を明らかにするため、研究員の現場の開発メンバー27名を対象にアンケートを実施した。アンケートでの質問とその回答からわかったことを以下に示す。

(1) どのような調査方法で調査を行なっているのか？

基本の構造を理解するために行なう調査や変更箇所を特定するための調査、変更による影響範囲を特定するための調査を一緒に行なっている人が多い。

(2) 調査結果をどのように記録しているか？

調査結果を文書化していない。または、文書化していても、記載内容が人によって様々である。記載すべき内容が決まっていないため、記載すべきことに抜け漏れがある。

(3) 調査範囲を決めて調査を行なっているか？

調査範囲を明確に設定せずに、調査を開始している人が多い。

(4) 調査において、ムダだと感じている点はあるか？

調査のたびに、同じ箇所を調査した経験のある人が多い。

(5) 調査不備による不具合の発生経験の有無は？

約60%の人が経験ありと回答しており、半数以上を占めている。

以上のことから、我々の派生開発の現場における影響範囲の調査では、方法が個人に依存しており個人の力量に任せていることがわかった。このことが「調査方法の問題」に起因する不具合を引き起こしていると考えられる。

2.3 先行研究

変更の影響を受ける範囲を特定する調査プロセスの問題を解決するために、開発プロセスである XDDP[1][2][3]の技術や先行研究[4]が参考にならないか確認した。

まず、XDDPでは、変更箇所の調査において適当な設計資料が残されていない場合、“スペックアウト”と呼ばれる方法でソースコードを調査する過程で調査資料を残し、見つけた変更箇所は変更要求仕様書に書き出して残している。これにより、ソースコードを探す過程で該当する箇所を見つけ次第に変更してしまう状況を回避できている。しかし、調査資料の残し方やその内容に関しては、書籍等でも紹介するに留めている。

次に、大規模、複雑化したソフトウェアでの変更の影響を受ける箇所の見落としを防止し、影響を受ける箇所の調査の属人性を低減する手法がある[4]。これは階層化したトレーサビリティマトリクスを活用し、影響を受ける箇所を表現する方法と変更パターンごとの影響を受ける箇所の調査方法が論じられている。この手法は、変更要求・テスト項目・ソフトウェア構造の関係を可視化する手法であり、調査プロセスの実施タイミングが異なるため適用することが困難と判断した。

上記の結果から、調査プロセスの問題を解決する方法は書籍等で紹介されているものの、現場で適用できるほど具体化はされていない。そこで、我々は、調査の問題にフォーカスを絞り、現場で活用できる具体的な施策として、変更による影響範囲の特定漏れを低減する調査プロセスについて検討した。

3. 解決策

3.1 初心者と熟練者と XDDP の調査プロセスのギャップ

前章で、担当者ごとに調査プロセスが異なっていることがわかった。調査プロセスの相違点を明かにするために、実務経験の少ない担当者（以降、初心者）、実務経験が多い担当者（以降、熟練者）と、派生開発で効果的な手法とされる XDDP の3つの調査プロセスを比較した。表1にそれぞれの調査プロセスのギャップを示す。

その結果、それぞれの調査プロセスには、調査目的、ソースコードの処理を探索するル

第6分科会 (Aグループ)

ート、調査資料の残し方、ソースコードの理解の仕方に違いがあることがわかった。特に、初心者は、ソースコードの構造を把握する前に変更箇所を探し、影響を受ける箇所の特定漏れを起こしやすい。また、調査の結果を残していない、あるいは成果物をレビューしても調査目的・内容・範囲などがわからず、調査の抜け漏れに気付きにくい。これらの問題を引き起こす原因が何かを分析したところ、ソースコードを調べる際の探索ルートが原因であることがわかった。

表 1 初心者、熟練者と XDDP の調査プロセスのギャップ

項目	初心者	熟練者	XDDP
変更要求	変更要求を理解するプロセスがない, 言われた言葉のまま調査を開始している	XDDP と同じ	変更要求を理解するプロセスがある
調査目的	調査目的が明確になっていないため, 構造を理解する調査, 変更箇所を特定する調査, 影響範囲を特定する調査を一緒に実施している	XDDP と同じ	調査目的が明確になっているため, 構造を理解する調査, 変更箇所を特定する調査, 影響範囲を特定する調査を分離させている
ソースコードの探索ルート	初心者は目についた順にソースを辿っている	XDDP と同じ	一つの関数を最後まで見て, 次の関数を見る
調査資料の残し方	メモ書き, 時には, 何も残っていないこともある	メモ書き, 場合によっては構造図, 何らかのチャート	構造図, PAD
ソースコードの理解の仕方	ソースコードを読みながら理解	ソースコードを読みながら理解	ソースコードから変換した構造図, PAD を理解(ソースコードを理解しない)

3.2 探索ルートの問題

ここでは, 3.1 で述べたソースコードの探索ルートの問題点を示す。図 2 は問題がある探索ルート B (破線) とそうでない探索ルート A (実線) を示している。

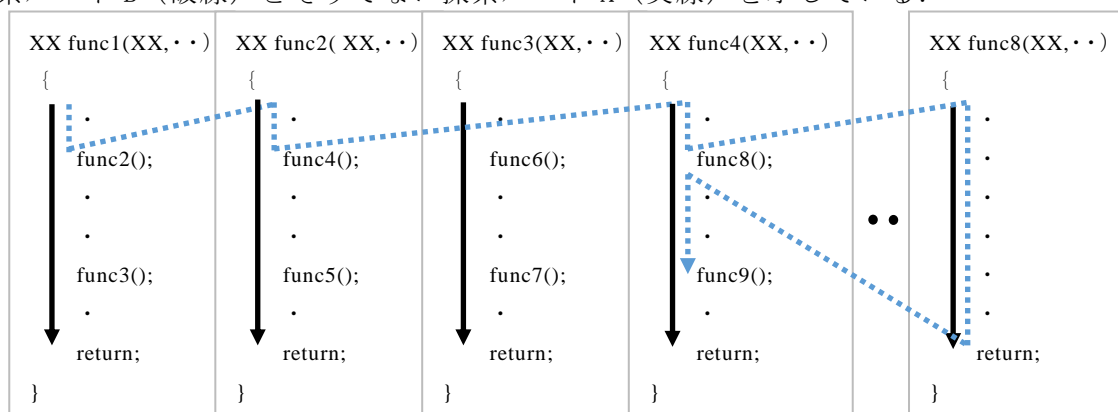


図 2 ソースコードの探索

探索ルート A は XDDP でも勧めている方法で, 調査している関数内に新たな関数が現れても, 個々の関数ごとに最後まで調査する。探索ルート B は, 関数が現れるたびにその関数の中に次々と入って調査を行なう。探索ルート A より影響範囲の特定漏れを起こしやすい探索ルート B の問題点 4 つを述べる。

第6分科会 (Aグループ)

(1) 問題点 1 (狭い範囲しか調べられないことによって起こる問題)

探索ルート A がソースコードの処理を構造「面」で捉えることに対して、探索ルート B は処理を順番「線」で捉える。このことから、探索ルート B の調査範囲は A に比べると狭く、他の変更箇所気付けないという問題に繋がる。調べた範囲が限定的であるため、より効果的な変更箇所が他にあることに気付けないこともある。また、複数の関数を往來するなど、ソースコードを辿ることに時間がかかり、制限時間内に最後まで探索できないことも起きてしまう。

(2) 問題点 2 (調査目的の違う調査を一緒に行なうことによって起こる問題)

変更箇所や方法によって影響を受ける箇所が変わるため、「変更箇所の特定」「影響を受ける箇所の特定」を順に実施する必要がある。探索ルート B のような調べ方では、目的別に調査を区別して実施できないため、変更箇所が変わり、それによる影響を受ける箇所も変わること気付かず、影響を受ける箇所を漏らすという問題が発生する。

(3) 問題点 3 (構造を把握しないことによる問題)

一般に、機能を実現するのに 10 数個から数 10 個の関数がそれぞれの役割を分担して組み立てられている。この状況で B のルートで探索すると、関数群の役割を把握できないため、変更に対する影響を受ける箇所を特定しづらい。

(4) 問題点 4 (調査結果についての問題)

探索ルート B では、調査している関数の中に新たな関数が現れると、その関数の中に入っていくため、図 4 のように辿った経路を記録したメモしか残らない。そのため調べた結果がレビュー時や次の派生開発時にも使えるような調査資料にならない。

一方、探索ルート A では、1 つの関数毎に処理内容を調査するので、図 3 の関数の構造図や PAD を残すことが容易である。調査を中断する場合も、調べ終わった関数を明確に切り分け、調べた範囲を表現でき、信頼性があり次にも利用できる資料となる。

このように、探索ルート A は B に比べて、変更箇所や影響範囲の特定漏れを起こしにくく、調査時間や範囲、調査結果の活用性・再利用性においても優れていることがわかった。

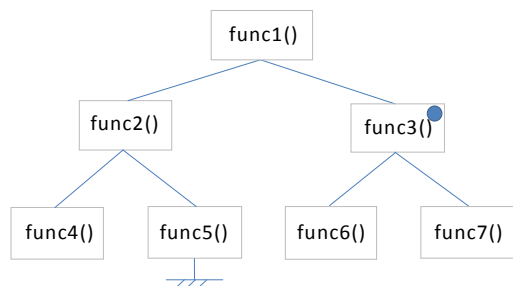


図 3 探索ルート A の成果物 (構造図)

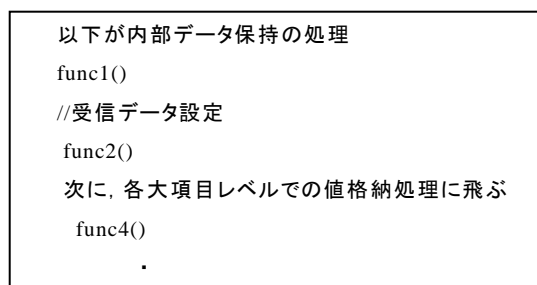


図 4 探索ルート B の成果物 (メモ)

3.3 標準調査プロセスの定義

3.2 の探索ルートの問題によって、影響範囲の特定漏れを起こしやすいことがわかった。この特定漏れの問題を低減するためにプロセスの統一および調査結果のテンプレート化 (レビューインプットの明確化) を行ない、変更の影響範囲を特定するための「標準調査プロセス」を定義した。現状、目的が違う 3 つの調査を一緒に行なうことで、影響範囲の特定漏れを起こしているため、調査ステージを目的別に 3 つに分類した。以下の調査ステージの内容を示す。現段階では、処理の流れなどの静的な構造の理解を目的としており、動的な振る舞い (例: 状態遷移、タイミングチャート) の理解については対応していない。

表2 標準調査プロセスの3つの調査ステージ

No	調査ステージ	説明
1	事前調査	依頼内容を理解するために実施する、予め知っておくべきシステムの構造や動作を把握する調査、調査範囲を決めて調査し、成果物を残す、この調査では変更箇所は探さない 構造図を作成する際には、1関数に対して大体1分程度の時間となる。その値と調査対象の関数の数と調査に使える工数から、何関数を調査できるかを計算し、調査範囲を決定する
2	変更箇所調査	変更要求を実現するために必要な、システムの変更箇所を特定する調査 (XDDPでいう「スペックアウト」に相当する) ソースコードを追って理解しながら変更箇所を探す ひとつおとり変更仕様が抽出できたら調査を終了する
3	影響調査	システムの変更によって、影響が及ぶ箇所を特定する調査 (XDDPでいう「スペックアウト」に相当する) 変更箇所があるとは限らない 変更箇所調査が完了した後に実施する、理由としては、変更方法や変更箇所によって影響の仕方が変わるため

このように、表2の調査ステージを明確にして、ソースコードを調査することで、担当者によって差があった調査方法が統一され、適切な調査ステージで適切な調査が行なえる。またレビューに耐えうる調査結果を残すことができ、調査の抜け漏れを防ぐことができる。

3.4 調査プロセスガイドラインの作成

プロセスの定義だけでは、実際の現場でどのように活動すれば良いか分かりづらい。そのため、「標準調査プロセス」をより具体的な活動指針として明文化した「調査プロセスガイドライン」を作成した。「調査プロセスガイドライン」には、熟練者の調査ノウハウをチェックポイントとして記載した。これにより初めて調査する人も「標準調査プロセス」に則って調査を行うことができ、変更による影響範囲の特定漏れを低減する効果が期待できる。さらに初心者育成の効果も期待できる。以下に「調査プロセスガイドライン」を用いた際の調査ステップの概要について説明する。ステップの詳細は、「調査プロセスガイドライン」の4.2.1章の調査プロセスPFDとプロセス定義書を参照する。調査担当者が初心者の場合は、「調査プロセスガイドライン」を十分に活用できるよう、事前にトレーニングすることを推奨する。

(1)ステップ1

付録の「調査プロセスガイドライン」の5章の調査プロセスガイドラインの適用方法に従い、変更要求の調査目的や調査内容に応じて、パターンを決定する。

- 【基本パターン1】調査ステージ「事前調査」「変更箇所調査」「影響調査」を実施
- ・変更依頼を受け、システム構造の理解や変更箇所を特定する場合の基本パターン
 - ・担当者が対象システムの知識や保守経験が浅い場合は、「事前調査」から始める

【基本パターン2】調査ステージ「変更箇所調査」「影響調査」を実施

- ・以下に該当する場合は、「事前調査」を省略できる
 担当者が変更箇所を特定できている、もしくは特定できるだけの資料がある
 担当者が構造や処理の流れ（現行仕様）を理解している

(2)ステップ2

基本パターン1の場合は、付録2の調査プロセスのテンプレート（調査テーマ記入シート）を記載する。ここでそれぞれの欄を記載したら、内容が妥当かどうかレビューを実施する。これによって、調査目的、調査範囲、調査時間が明確になり、事前調査が発散しないようになる。事前調査実施の欄は事前調査終了後に記載する。基本パターン2の場合はこのステップは実行しない。

(3)ステップ3

基本パターン1の場合は、「事前調査」「変更箇所調査」「影響調査」を実施する。

第6分科会 (Aグループ)

基本パターン2の場合は、「変更箇所調査」「影響調査」を実施する。その際に、付録1のプロセスチェックポイントと付録3の調査実施事例（探索ルートと成果物）を参考にす。プロセスチェックポイントには、調査プロセス PFD とプロセス定義書に対応した PFD-No に対するチェックポイントとアドバイスが記載されている。調査実施事例（探索ルートと成果物）には、XDDP の調査プロセスで用いられている探索の仕方（探索ルート A）、その調査結果の残し方、影響範囲の特定漏れを起ししやすい探索の仕方（探索ルート B）が記載されている。これらの資料を基に、ソースコードを効率よく漏れなく調査することによって、変更による影響範囲の特定漏れを低減し、有用な調査資料を残すことができる。

4. 解決策の検証

4.1 標準調査プロセスと調査プロセスガイドラインの検証方法

「標準調査プロセス」と「調査プロセスガイドライン」の有用性を判断するために、次の3点について検証した。なお、検証は過去の不具合事例に対して行なうものとする。ただし、「調査プロセスガイドライン」はドラフト版である。

- (1) 調査のステージを意識して調査を行なうことができるか
- (2) 不具合事例の原因となった影響を受ける範囲や変更箇所を特定できるか
- (3) メモ書きではない調査結果が残せるか

検証範囲は2.1で「調査方法の問題」に分類した不具合事例40件の内、変更による影響範囲が無知見領域に存在した事例2件、無知見領域ではないが、変更による影響範囲に気付かなかった事例3件を対象とした。また検証方法は、不具合が発生したきっかけとなった変更要求に対し、「標準調査プロセス」と「調査プロセスガイドライン」を用いて、シミュレーションを行ない、調査結果を残すまでを実施するという方法で行なった。

4.2 標準調査プロセスと調査プロセスガイドラインの有用性の検証結果

5件の不具合についてシミュレーションした結果を表3に示す。

表3 標準調査プロセスと調査プロセスガイドラインの有用性の検証結果

不具合事例	変更の影響範囲	調査ステージ	結果	成果物	調査工数	構造図作成工数
事例1	無知見	事前調査	特定できた	構造図	5時間	5時間 (64関数)
事例2	無知見	事前調査	特定できた	構造図	4時間	1.4時間 (70関数)
事例3	知見	事前調査 変更箇所調査	特定できた	構造図 フローチャート	3時間	1.8時間 (104関数)
事例4	知見	変更箇所調査	特定できた	フローチャート	2時間	作成していない
事例5	知見	事前調査 変更箇所調査	特定できなかった	構造図 フローチャート	1時間	0.5時間 (28関数)

表中の「変更の影響範囲」は無知見領域に存在したか、知見領域に存在したかを意味する。また「調査ステージ」にはどの調査ステージを実施したかを記載する。

影響範囲を特定できた4件の事例については、「事前調査」ステージで基本的な構造や動作を理解するための調査を行ない、成果物として構造図やフローチャートを残せた。無知見領域に影響範囲が存在した事例では、これらの調査結果から影響を受ける箇所を特定することができた。知見領域に影響範囲が存在した事例では、調査資料を基に「変更箇所調査」ステージで行なった調査において、変更箇所を特定することができた。

特定できなかった事例では、特定できた事例と同様に構造図やフローチャートを残せた

第6分科会（Aグループ）

が、変更箇所を特定できなかった。

5つの事例を基に、構造図に記載した関数と構造図の作成時間から、1時間あたり構造図に約30関数を記載できることがわかった。ただし、フローチャートの作成時間は除く。

4.3 標準調査プロセスと調査プロセスガイドラインの有用性についての考察

検証の結果、調査のステージを意識して調査を行なうことができたことから、混同されていた構造を理解する調査、変更箇所を特定する調査、影響範囲を特定する調査を分けて行なうことができ、混同していることで気付かなかった変更による影響範囲の特定漏れを低減することができると考えられる。また、メモ書きでない調査結果を残すことができたことから、レビューのインプットとなることがいえる。これらのことから、「標準調査プロセス」と「調査プロセスガイドライン」は変更による影響範囲の特定漏れの低減に効果があるといえる。さらに、構造図を作成する時間がわかったため、これを事前調査の見積りに使うことができると考える。しかし、表3の事例5では、変更による影響範囲を特定できなかった。事例5はイベントドリブンの関数内に影響を受ける箇所が存在し、イベントの順序とイベントを受けたときの状態によって、振る舞いが変わっていた。これは動的な振る舞いのケースであると推測されるため、今回の「標準調査プロセス」と「調査プロセスガイドライン」の適用範囲外となるため、特定できなかったと考えられる。また、シミュレーションは研究員が行っており、現場で有効に活用できるかは検証できていない。

5. 本研究のまとめ

5.1 研究成果

今回の研究を通し、調査プロセスの問題を分析したところ、構造を理解するための調査、変更箇所を探すための調査、影響を受ける箇所を探すための調査を混同して行っていたために目的に合わない調査を行っていることに気付いた。また、調査結果としてメモ書きしか残せておらず、レビューによって抜け漏れに気づきづらくなっていることがわかった。解決法として「標準調査プロセス」と「調査プロセスガイドライン」を作成し、検証した結果、以下の効果があることがわかった。

- ・適切な調査のステージで適切な調査を行なえる
- ・処理の流れなど静的なケースにおける、変更による影響範囲の特定漏れを低減できる
- ・調査結果を残すことができ、レビューが可能となる

5.2 今後の進め方

我々の考案した「標準調査プロセス」と「調査プロセスガイドライン」は、実際に発生した不具合事例に対するシミュレーションにより効果があることがわかったが、動的な振る舞いの調査方法については着目できていない。今後は、実際のプロジェクトで運用し有効性を高め、また動的な振る舞いの理解を目的とする調査方法についても着目し、「標準調査プロセス」と「調査プロセスガイドライン」を改良していきたい。

6. 参考文献

- [1] 清水 吉男, “「派生開発」を成功させるプロセス改善の技術と極意”, 2007 年
- [2] 清水 吉男, “[入門+実践]要求を仕様化する技術・表現する技術 -仕様が書けていますか?”, 2005 年
- [3] 清水 吉男, “派生開発プロセス[XDDP]のポイント -XDDP の考え方を知る-”, SQiP 第 6 分科会[派生開発]特別講演資料, 2014
- [4] ET-WEST2014 AFFORDD セッション資料, “派生開発における影響箇所の気づき “

2015年1月
第1.0版

変更の影響範囲を特定するための 調査プロセスガイドライン

第30年度（2014年度）SQiP研究会
第6分科会Aグループ

目次

1. 目的	3
2. 適用範囲	3
3. 「標準調査プロセス」及び「調査プロセスガイドライン」の重要性について	3
4. 標準調査プロセス	3
4.1 標準調査プロセスの目的	3
4.2 標準調査プロセスの定義	4
5. 調査プロセスガイドラインの適用方法	9
付録1： プロセスチェックポイント	10
付録2： 調査プロセスのテンプレート（調査テーマ記入シート）	11
付録3： 調査実施事例（探索ルートと成果物）	13

PFD : Process Flow Diagram の略. 合理的なプロセスを設計するためのツールで清水吉男氏がDFDをアレンジして考案したものです. PFDの書き方については硬派のホームページの「PFDの書き方」を参照してください. (http://homepage3.nifty.com/koha_hp/process/PFDform3.pdf)

1. 目的

このガイドラインは、「ソースコード調査時の考慮漏れ」や「影響範囲の特定漏れ」といった不具合の発生を抑えるために定義された変更による影響範囲を特定する「標準調査プロセス」を調査内容に応じて効果的に活用する方法をガイドすることを目的とする。

2. 適用範囲

このガイドラインは、ソフトウェアの派生開発を実施している組織又はプロジェクトにおいて、ソースコードや設計書を基に、処理の流れなど静的な構造の理解を目的とする調査に適用する。

注) 本ガイドで提供する標準調査プロセスでは、動的な振る舞い（例：状態遷移、タイミングチャート）の理解を目的とする調査には、現時点では対応していない。

3. 「標準調査プロセス」及び「調査プロセスガイドライン」の重要性について

派生開発などの変更を伴うシステム開発では、設計工程よりも前に、変更箇所、及び、変更による影響範囲を特定するための調査が必要である。特に、頻繁に発生する変更要求に対処するためには、限られた調査時間内に、この調査プロセスをいかに効率良く、精度よく実施できるかが、品質・コスト・納期(QCD)の目標達成、ひいてはビジネスの成否をも左右する。

しかしながら、現場でどのような方法を用いて変更による影響範囲を特定しているのか調査したところ、変更箇所を探す調査とそこから影響を受けて変更が必要となる箇所を探す調査（＝影響調査）を同時に実行していることが分かった。また、担当者によって調査方法は異なり、調査結果の残し方も様々であることから、結果として期待ほどの成果が得られていない（特に、調査経験の浅い技術者に多く見られる）ことが分かった。

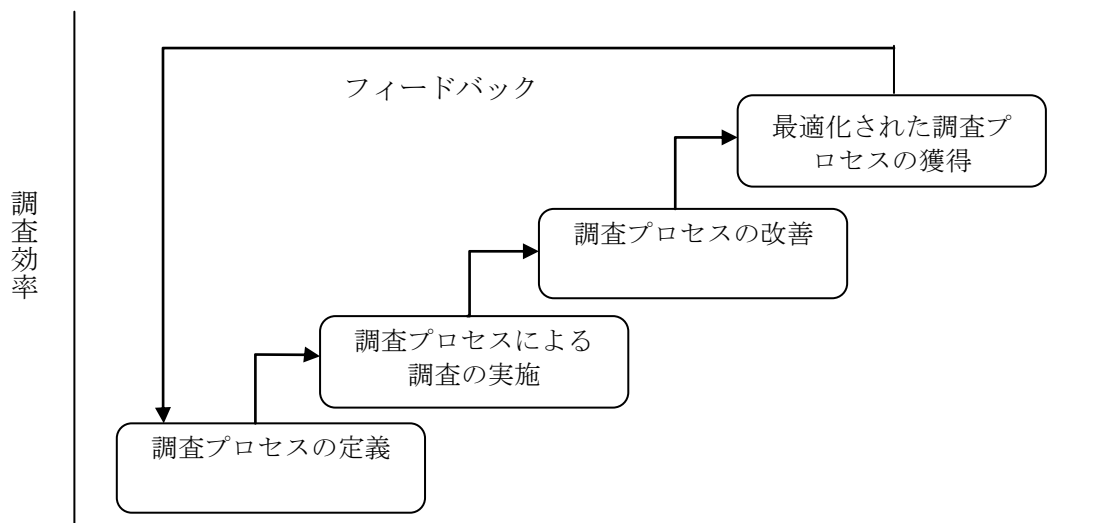
そこで、調査方法及び調査結果の人によるバラツキ（属人性）を排除し、熟練者のノウハウを盛り込むことを目的として、変更の影響範囲を特定するための「標準調査プロセス」を定義した。さらに「標準調査プロセス」を調査内容に応じて効果的に活用する方法を「調査プロセスガイドライン」としてまとめた。

4. 標準調査プロセス

4.1 標準調査プロセスの目的

標準調査プロセスの目的は、調査方法及び調査結果の人によるバラツキ（属人性）を排除し、熟練者のノウハウを盛り込むことで、変更の影響範囲を引き起こす「ソースコード調査時の考慮漏れ」や「影響範囲の特定漏れ」といった不具合の発生を抑えることを目的とする。

なお、調査プロセスを標準化することで、図1に示すような調査プロセスの改善サイクルを回すことを可能とする。



4.2 標準調査プロセスの定義

標準調査プロセスは、表1に示すように、大きく「事前調査」、「変更箇所調査」、「影響調査」の3つの調査ステージに区分する。これらの調査ステージには、調査を始める前に必要な「変更要求の理解」と「調査計画」を含む。「変更要求の理解」は事前調査の初期段階で実施し、「調査計画」は各調査ステージで実施する。

表1 標準調査プロセスの3つの調査ステージ

No	調査ステージ	説明
1	事前調査	依頼内容を理解するために実施する、予め知っておくべきシステムの構造や動作を把握する調査。調査範囲を決めて調査し、成果物を残す。この調査では変更箇所は探さない。 構造図を作成する際には、1関数に対して大体1分程度の時間となる。その値と調査対象の関数の数と調査に使える工数から、何関数を調査できるかを計算し、調査範囲を決定する。
2	変更箇所調査	変更要求を実現するために必要な、システムの変更箇所を特定する調査（XDDPでいう「スペックアウト」に相当する）。 ソースコードを追って理解しながら変更箇所を探す。 ひとつおとり変更仕様が抽出できたら調査を終了する。
3	影響調査	システムの変更によって、影響が及ぶ箇所を特定する調査。 （XDDPでいう「スペックアウト」に相当する） 変更箇所があるとは限らない 変更箇所調査が完了した後に実施する。理由としては、変更方法や変更箇所によって影響の仕方が変わるため。

4.2.1 調査プロセス PFD とプロセス定義書

プロセス定義書では、調査ステージごとの作業内容、入力情報、出力情報などについて定義している。また、それぞれの関係をプロセスフロー図（ProcessFlowDiagram）に示している。これらを基に、各調査ステージで具体的に実施する作業内容や必要な情報の洗い出しを行なう。

■調査プロセス PFD

図 1-A. 事前調査プロセスの PFD

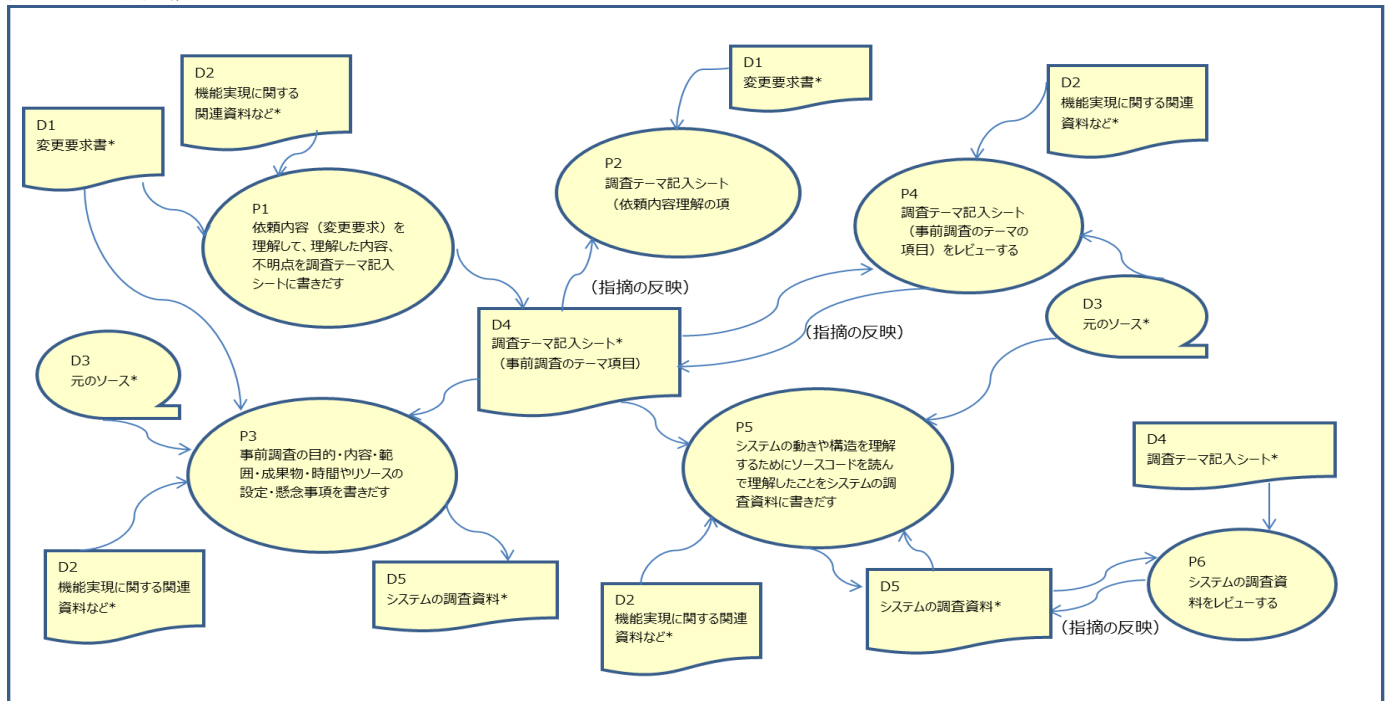


図 1-B. 変更箇所調査プロセスの PFD

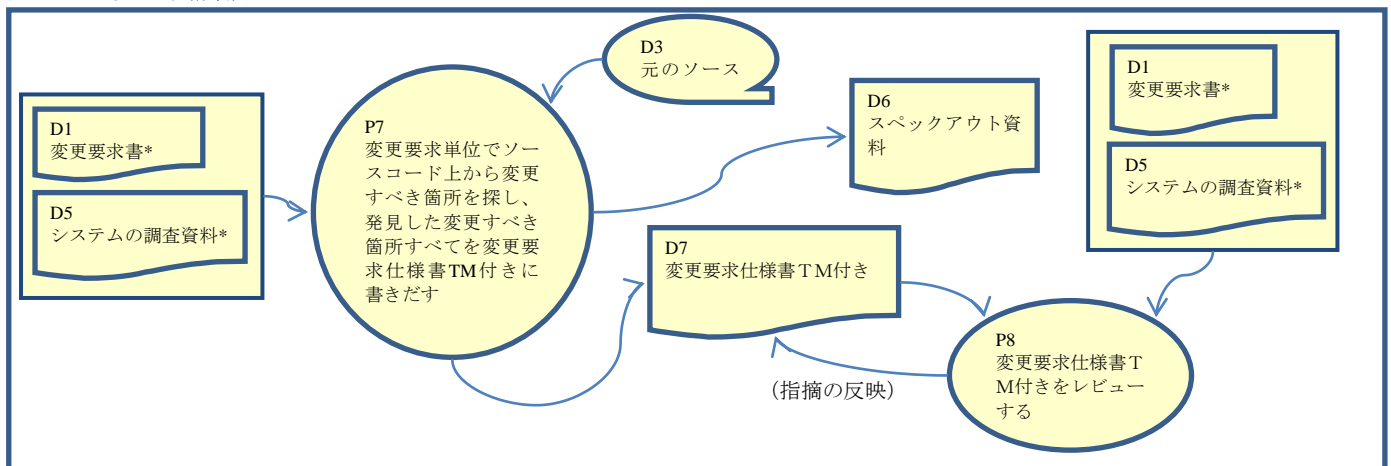
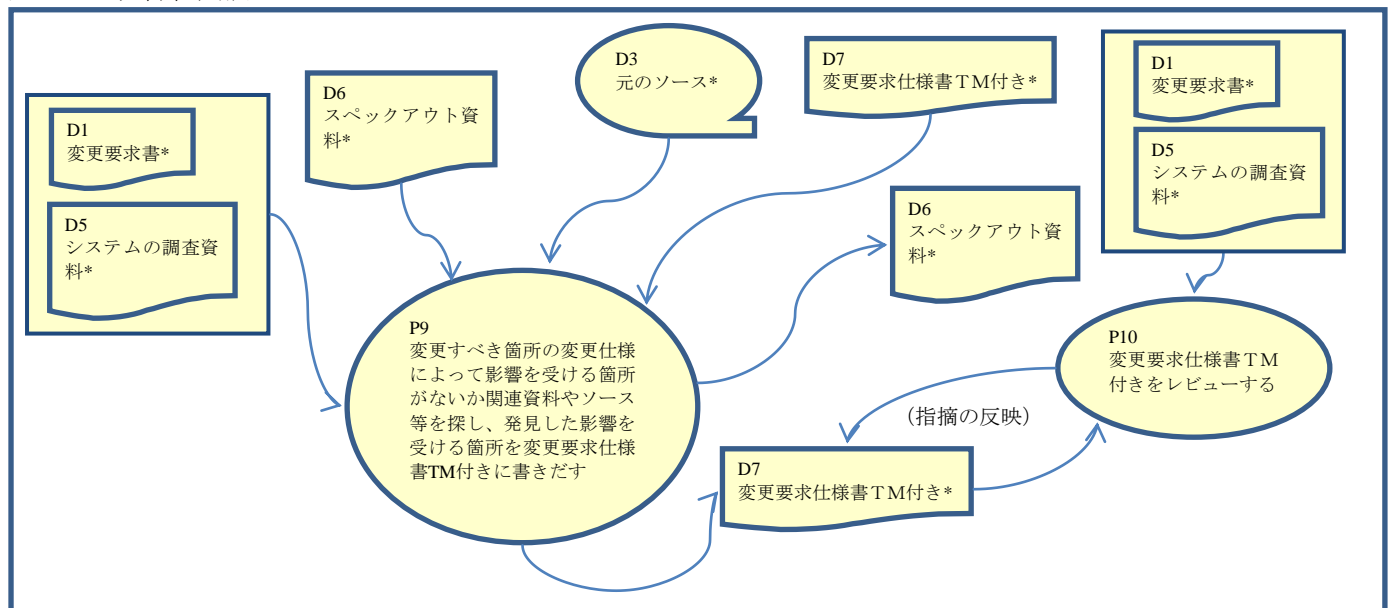


図 1-C. 影響範囲調査プロセスの PFD



■プロセス定義書

プロセス番号		プロセス名		作成日
P1		依頼内容（変更要求）を理解して、理解した内容、不明点を調査テーマ記入シートに書きだす		
入力情報	D1	変更要求書		
出力情報	D4	調査テーマ記入シート（依頼内容理解）		
作業内容				作成担当者
プロセス実施条件	変更要求書が届いた時点			
必要スキル				
担当者				
作業内容	1	依頼内容（変更要求）を調査テーマ記入シート（依頼内容）【D4】に書き出す		
	2	依頼内容（変更要求）から理解した内容を調査テーマ記入シート（依頼内容理解）【D4】に書き出す		
	3	依頼内容（変更要求）の不明点を調査テーマ記入シート（依頼内容理解）【D4】に書き出す		
プロセス終了判定	調査テーマ記入シート（依頼内容理解）が作成出来た状態			

プロセス番号		プロセス名		作成日
P2		調査テーマ記入シート（依頼内容理解）をレビューする		
入力情報	D1	変更要求書		
	D4	調査テーマ記入シート（依頼内容理解）		
出力情報	D4	調査テーマ記入シート（依頼内容理解、指摘内容）		
作業内容				作成担当者
プロセス実施条件	調査テーマ記入シート（依頼内容理解）が作成できた時点			
必要スキル				
担当者				
作業内容	1	依頼内容（変更要求）から理解した内容が、顧客の要求にあった内容か確認し、指摘事項を調査テーマ記入シート（指摘内容の欄）【D4】に記載する		
	2	依頼内容（変更要求）の不明点に対する回答を調査テーマ記入シート（指摘内容の欄）【D4】に記載する		
	3	レビュー指摘に対する修正内容を調査テーマ記入シート（依頼内容の理解）【D4】に記載する		
プロセス終了判定	調査テーマ記入シート（依頼内容理解の項目）のレビューが完了し、指摘の反映、反映内容の確認が完了した状態			

プロセス番号		プロセス名		作成日
P3		事前調査の目的・内容・範囲・成果物・時間やリソースの設定・懸念事項を書きだす		
入力情報	D2	機能実現に関する関連資料など		
	D3	元のソース		
	D4	調査テーマ記入シート（依頼内容理解の項目）		
出力情報	D4	調査テーマ記入シート（事前調査テーマの項目）		
	D5	システムの調査資料		
作業内容				作成担当者
プロセス実施条件	レビュー済みの調査テーマ記入シート（依頼内容理解の項目）が作成できた時点			
必要スキル				
担当者				
作業内容	1	依頼内容から関係する機能仕様書や各種の設計書、関数仕様書などの文書を集めて一覧にし、システムの調査資料【D5】に記載する		
	2	ソースファイルの一覧を作成し、システムの調査資料【D5】に記載する。またソースファイルが扱っている機能や所属するタスクなどの情報を書き込む		
	3	調査目的を明確にし、調査テーマ記入シート（事前調査テーマの項目）【D4】に記載する（調査目的）		
	4	最初に理解しなければならない（と思われる）「基本的な動き」を具体的にリストアップし、調査テーマ記入シート（事前調査テーマの項目）【D4】に記載する（調査内容）		
	5	何階層まで調査するのかの決め方の一つとして、スペックアウト時間（1関数×1分）（*1）から範囲を求めることができる。（*1）スペックアウト時間は過去の実績値がある場合はそちらを使用すること。		
	6	スペックアウト資料を残す、など成果物を決め、調査テーマ記入シート（事前調査テーマの項目）【D4】に記載する（調査の成果物）		
	7	スペックアウトの時間などを参考に、文書・ソース一覧の情報から見積もり、調査テーマ記入シート（事前調査テーマの項目）【D4】に記載する（時間やリソースの設定）		
	8	懸念事項がないか確認し、調査テーマ記入シート（事前調査テーマの項目）【D4】に記載する（懸念事項）		
プロセス終了判定	調査テーマ記入シート（事前調査テーマの項目）が作成出来た状態			

付録 調査プロセスガイドライン

プロセス番号		プロセス名		作成日
P4		調査テーマ記入シート（事前調査テーマ）をレビューする		
入力情報	D2 D3 D4	機能実現に関する関連資料など 元のソース 調査テーマ記入シート（依頼内容理解）		
出力情報	D4	調査テーマ記入シート（事前調査テーマ、指摘内容）		
作業内容				作成担当者
プロセス実施条件	調査テーマ記入シート（事前調査テーマ）が作成できた時点			
必要スキル				
担当者				
作業内容	1	調査のテーマを以下の観点でレビューし、指摘事項を調査テーマ記入シート（指摘内容の欄）【D4】に記載する ・調査目的が事前調査で明らかにしたい事柄とあっているか（調査目的） ・調査内容に抜けはないか、他に理解しなければならない項目はないか（調査内容） ・関数の呼び出し階層が少なくないか、多くないか（調査範囲） ・基本的な動きを理解するために必要な成果物か（調査の成果物） ・調査時間やリソースは適切か（時間やリソースの設定） ・他に懸念事項はないか、本当に懸念事項かどうか（懸念事項）		
	2	レビュー指摘に対する修正内容を調査テーマ記入シート（事前調査テーマ）【D4】に記載する		
プロセス終了判定	調査テーマ記入シート（事前調査テーマ）のレビューが完了し、指摘反映、反映内容の確認が完了した状態			

プロセス番号		プロセス名		作成日
P5		システムの動きや構造を理解するためにソースコードを読んで理解したことをシステムの調査資料に書き出す		
入力情報	D2 D3 D4 D5	機能実現に関する関連資料など元のソース調査テーマ記入シートシステムの調査資料		
出力情報	D5	システムの調査資料		
作業内容				作成担当者
プロセス実施条件	レビュー済みの調査テーマ記入シートが作成できた時点			
必要スキル				
担当者				
作業内容	1	全体（ある範囲）の構造や動作する仕組みを理解するために「ソースコードを読んで理解（＝スペックアウト）」し、理解したことをシステムの調査資料【D5】に記載する		
	2	システム設計書やタスク設計書を補うようなシステムの構成や関係を表す情報をシステムの調査資料【D5】に記載する		
プロセス終了判定	調査テーマ記入シートに沿った調査結果資料が作成出来た状態			

プロセス番号		プロセス名		作成日
P6		システムの調査資料をレビューする		
入力情報	D4 D5	調査テーマ記入シート システムの調査資料		
出力情報	D5	システムの調査資料（指摘内容込み）		
作業内容				作成担当者
プロセス実施条件	調査テーマ記入シート（事前調査テーマ）が作成できた時点			
必要スキル				
担当者				
作業内容	1	システムの調査資料を以下の観点でレビューし、指摘内容をシステムの調査資料【D5】に記載する ・調査内容に抜けはないか ・調査内容が理解できるか		
	2	レビュー指摘に対する修正内容をシステムの調査資料【D5】に記載する		
プロセス終了判定	システムの調査資料のレビューが完了し、指摘の反映、反映内容の確認が完了した状態			

プロセス番号		プロセス名		作成日
P7		変更要求単位でソースコード上から変更すべき箇所を探し、発見した変更すべき箇所すべてを変更要求仕様書 TM 付きに書き出す		
入力情報	D1 D3 D5	変更要求書 元のソース システムの調査資料		
出力情報	D6 D7	スペックアウト資料 変更要求仕様書 TM 付き		
作業内容				作成担当者
プロセス実施条件	変更要求書が届いた時点 あるいは事前調査が完了した時点			
必要スキル				
担当者				
作業内容	1	変更要求単位で、ソースコード上で変更すべき箇所の検討をつけ、変更要求仕様書 TM 付き【D7】に記載する。あるいは機能仕様書からソースコード上の変更箇所を特定できるレベルの変更仕様が抽出されている場合は、その変更箇所を、変更要求仕様書 TM 付き【D7】に記載する。変更箇所の検討がつかない場合は、関連しそうなソースコードを読んで、構造図、PADなどの資料【D6】を作成する。スペックアウト資料【D6】の内容を理解し、変更箇所を特定できたら、発見した変更箇所（変更仕様）をすべて変更仕様書 TM 付き【D7】に記載する。		
プロセス終了判定	変更すべき箇所を変更要求仕様書 TM 付きに記載出来た状態 スペックアウト資料が必要な場合、スペックアウト資料が作成できた状態			

付録 調査プロセスガイドライン

プロセス番号		プロセス名		作成日
P8		変更要求仕様書TM付きをレビューする		
入力情報	D1	変更要求書		
	D5	システムの調査資料		
出力情報	D6	スペックアウト資料		
	D7	変更要求仕様書 TM 付き		
作業内容				作成担当者
プロセス実施条件		変更要求仕様書TM付きとあればスペックアウト資料が作成できた時点		
必要スキル				
担当者				
作業内容	1	変更要求仕様書TM付きを以下の観点でレビューする		
	2	<ul style="list-style-type: none"> 変更要求に対して、変更すべき箇所が適切に変更仕様として抽出されているか レビュー指摘に対する修正内容を変更要求仕様書 TM 付き【D7】に記載する 		
プロセス終了判定		変更要求仕様書TM付きのレビューが完了し、指摘の反映、反映内容の確認が完了した状態		
プロセス番号		プロセス名		作成日
P9		変更すべき箇所の変更仕様によって影響を受ける箇所がないか関連資料やソース等を探し、発見した影響を受ける箇所を変更要求仕様書 TM 付きに書きだす		
入力情報	D1	変更要求書		
	D3	元のソース		
出力情報	D5	システムの調査資料		
	D6	スペックアウト資料		
作業内容	D7	変更要求仕様書 TM 付き		作成担当者
	D6	スペックアウト資料		
プロセス実施条件		レビュー済みの変更箇所を特定した、変更要求仕様書TM付きが作成できた時点		
必要スキル				
担当者				
作業内容	1	変更箇所特定プロセスで発見した変更仕様によって影響を受ける箇所がないか関連資料やソース等から調べ、発見した影響を受ける箇所を変更要求仕様書 TM 付き【D7】に記載する。		
	2	影響を受ける箇所の必要な情報が関連文書等から得られない場合は、影響を受ける箇所のソースコードを読んで、構造図、PADなどの資料（スペックアウト資料【D6】）を作成する。スペックアウト資料【D6】の内容を理解し、変更箇所を特定できたら、変更仕様書 TM 付き【D7】に記載する。		
プロセス終了判定		変更すべき箇所を変更要求仕様書 TM 付きに記載出来た状態 スペックアウト資料が必要な場合、スペックアウト資料が作成できた状態		
プロセス番号		プロセス名		作成日
P10		変更要求仕様書TM付きをレビューする		
入力情報	D1	変更要求書		
	D5	システムの調査資料		
出力情報	D6	スペックアウト資料		
	D7	変更要求仕様書 TM 付き		
作業内容				作成担当者
プロセス実施条件		変更要求仕様書TM付きとあればスペックアウト資料が作成できた時点		
必要スキル				
担当者				
作業内容	1	変更要求仕様書TM付きを以下の観点でレビューする		
	2	<ul style="list-style-type: none"> 変更要求に対して、変更する箇所がほかにないかどうか レビュー指摘に対する修正内容を変更要求仕様書 TM 付き【D7】に記載する 		
プロセス終了判定		変更要求仕様書TM付きのレビューが完了し、指摘の反映、反映内容の確認が完了した状態		

5. 調査プロセスガイドラインの適用方法

変更要求の調査目的や調査内容によっては、3つの調査ステージをいつも全て実施する必要はない。

以下、標準調査プロセスの使い方の基本的なパターンを定義する。変更要求の調査目的や調査内容に応じて、基本パターンを使い分け、必要に応じて修整（テーラリング）して活用する。

<基本パターン>

【基本パターン1】 調査ステージ「事前調査」「変更箇所調査」「影響調査」を実施

- ・変更依頼を受け、システム構造の理解や変更箇所を特定する場合の基本的なパターン
- ・担当者が対象システムに関する知識や保守経験の浅い場合は、原則「事前調査」から始めること

【基本パターン2】 調査ステージ「変更箇所調査」「影響調査」を実施

- ・以下に該当する場合は、「事前調査」を省略できる
担当者が変更箇所を特定できている、もしくは特定できるだけの資料がある
担当者が構造や処理の流れ（現行仕様）を理解している

<標準調査プロセスを遂行するためのツール>

標準調査プロセスを遂行するために必要な、具体的なステップや情報をツールとしてまとめた。

(1). チェックポイント【付録1】

各調査ステージにおける詳細なチェックポイントやアドバイスについてまとめている。調査目的や内容は混合して作業しがちなため、チェックポイントシートを活用し、作業内容に見落としがないか確認する。

(2). 調査テーマ記入シート【付録2】

調査におけるタスクを調査ステージごとに分け、計画的に実施するためのテンプレートをまとめている。調査スケジュールを明確にし、進捗管理を行なう。

(3). 調査事例【付録3】

参考に、ソースコードからの処理の流れや構造を理解しようとする際の調査事例をまとめている。

付録 1： プロセスチェックポイント

PFD-No		チェックポイント、アドバイス
事前調査	P1	<p>依頼内容（変更要求）から理解した内容や不明点を調査テーマ記入シートに書き出す</p> <p>1 変更依頼から理解した内容や不明点を書き出す</p> <p>2 依頼者から求められている変更の意図をすべて、かつ適切に表現できたか確認する</p> <p>3 なぜ変更するのか、変更要求の理由を理解できているか確認する ※変更依頼の解釈を間違えると、設計間違いや変更範囲の特定漏れに繋がるので注意する！ ※「～だと思ふ」は個人的な解釈のため、依頼者に認識があつているかを確認する</p>
	P2	<p>調査テーマ記入シート（項目：依頼内容理解）をレビューする</p> <p>4 依頼内容を分類・整理できていることを、レビューによって検証する ・依頼内容の捉え方を間違っていないか ・キーワードに着目するあまり始めから調査範囲を狭めていないか など、依頼内容の誤った解釈による変更漏れを防ぐ</p>
	P3	<p>事前調査の目的・内容・成果物・時間やリソースの設定・懸念事項を書き出す</p> <p>5 依頼内容（変更要求）をもとに、自身の不得意領域（無知見領域）を確認する</p> <p>6 「システムの動きや構造を理解する」ために、実施すべき調査の内容を明確にする ※調査のテーマを決め、計画的に調査することで、ムダな調査を減らす</p>
	P4	<p>調査テーマ記入シート（項目：事前調査テーマ）をレビューする</p> <p>7 事前に調査すべき内容を理解できていることを、レビューによって検証する ※調査範囲を明確にし、 ・調査範囲の妥当性を確認する ・不要な調査範囲を削減する など 間違つた調査に時間をかけることを防ぐ</p>
	P5	<p>システムの動きや構造を理解するためにソースコードを読んで、理解したことをシステムの調査資料に書き出す</p> <p>8 事前調査テーマの項目と調査範囲にそつて、以下に示す調査を行なう ・システムのアーキテクチャ（構造）を確認する ・ベースのプログラムがどのような構造になっているか ・どのような仕掛けでデータが処理されているのか ※調査の仕方については【付録 3】調査事例を参照</p> <p>9 調べた結果の「事実」を記録する ※再利用性を高めるため、その時感じたことをコメントとして残しておくが良い ※調査結果の残し方については、【付録 3】調査事例を参照</p>
	P6	<p>システムの調査資料をレビューする</p> <p>10 システムの動きや構造が理解できていることを、レビューによって検証する ・本当に理解できているか ・思い込みや勘違いはないか など システム構造の誤った解釈による変更範囲特定漏れを防ぐ</p>
変更箇所調査	P7	<p>変更要求単位でソースコード上から変更すべき箇所を探し、発見した変更すべき箇所すべてを変更要求仕様書（TM付き）に書き出す</p> <p>11 変更仕様に対応するソースコードの変更箇所を掴む ※プログラム構造を理解できている場合、機能仕様書などの文書からソースコードの変更箇所を特定できることがある</p> <p>12 ソースコードを読みながら変更箇所を抽出する場合、 ・基本的に変更要求単位でソースコードを読む ・理解したあとで処理を書き出すのではなく、成果物に形を変えながら処理を理解すること ※中途半端な理解で書き出したものは信頼性に欠けるが、処理そのものを書き出したものは信頼性が高く、後に仕様書として活用できる</p> <p>13 ソースコードで設計を調べ、さらに詳しい変更仕様を探すために読む</p> <p>14 ソースコードを調べて発見した変更箇所は「変更仕様」の候補となり、変更要求の下に記載する ※「全検索」でのサーチ（Grep）の注意点は、変更箇所を”見つけ過ぎる”ことである 同じ変数を使つていても、必ずしもそこが変更箇所とは限らない</p> <p>15 ソースコードを調べた結果から変更仕様を調整する</p>
	P8	<p>変更要求仕様書（TM付き）をレビューする</p> <p>16 変更箇所の妥当性を、レビューによって検証する ・最初に発見した変更箇所の他に、効果的な変更箇所が隠れていないか ・思い込みで影響範囲を狭めていないか など 変更箇所の対応事項を確認する</p>
	P9	<p>ある仕様変更によって影響を受ける箇所を関連資料やソース等から探し出し、特定した影響を受ける箇所を変更要求仕様書（TM付き）に書き出す</p> <p>17 変更することで影響を受ける箇所を探す ※データ量や処理時間などのリソース変化による影響にも注意する ※変更が及ぼす影響を想像・推理しながら探す ※影響範囲を漏らさないために、 ・変更仕様をできるだけ具体的な表現で記述する ・関連する機能やパラメータに気付かせてくれる「補助資料」を活用する</p> <p>18 影響調査時に新しい変更箇所が見つかった場合は、変更仕様 TM に追記する</p>
影響調査	P10	<p>変更要求仕様書 TM 付きをレビューする</p> <p>19 変更による影響を受ける箇所をすべて洗い出せたか、レビューによって検証する ・影響を受ける箇所特定漏れを軽減する</p>

付録 2： 調査プロセスのテンプレート（調査テーマ記入シート）

■ 依頼事項

依頼件名			
依頼内容	担当者		問合せ日
依頼内容の理解/不明点			
指摘内容	レビューア		レビュー日
事前調査テーマ	<ul style="list-style-type: none"> ➢ 調査目的 ➢ 調査内容 ➢ 調査範囲 ➢ 調査の成果物 ➢ 時間やリソースの設定 ➢ 懸念事項 		
指摘内容	レビューア		レビュー日

■ 事前調査実施

実施結果	実施者		実施日		別紙	有り/無し
指摘内容	レビューア		レビュー日			

付録 2： 調査プロセスのテンプレート記載例（調査テーマ記入シート）

記載のポイント、レビューのポイントは
付録 1 プロセスチェックポイントを参考にする

■ 依頼事項

依頼件名 XXX 機能追加					
依頼内容	担当者	山田太郎	問合せ日	2015/AA/BB	
	XXX 機能追加に伴い、YYY モジュールも機能対応してほしい。				
依頼内容の理解/不明点	YYY モジュールに送信される以下のデータに変更がある ・ A のデータに新たに B のパラメータが追加される ・ B のデータに新たに C のパラメータが追加される また新たに D のデータが YYY モジュールに送信される。			調査担当者が変更依頼を受け取り理解した時点で記載	
指摘内容	レビューア	山田花子	レビュー日	2015/AA/CC	
	D のデータから YYY モジュールは E のデータを ZZZ モジュールに送信する必要がある。データ受信部だけでなく、データ送信部も含め、広範囲で調査する必要がある。YYY モジュール構造設計書は残っているが、古い資料のため、ソースコードと一致しない箇所が多数あると思われる。				
事前調査テーマ	▶ 調査目的 YYY モジュールは完全無知見のため、YYY モジュールの構造、処理を理解する。				熟練者が担当者の理解した内容をレビューし記載
	▶ 調査内容 YYY モジュールの構造、処理を把握する。				調査担当者が事前調査実施前に記載
	▶ 調査範囲 調査時間は 10H。1 関数を構造図に記載する時間は 1min なので 5H を使い main 関数から 300 関数を調べ、構造図を記載する。残り 5H で処理が不明な関数の PAD を起こす。				目安として、1 関数を構造図に記載する時間は 1min。構造図を作成する時間 = 1min × 調査関数の数
	▶ 調査の成果物 構造図、PAD(処理がわかりづらい場合)				
	▶ 時間やリソースの設定 担当 1 名(山田太郎) 調査時間 10H				
	▶ 懸念事項 YYY モジュール構造設計書はメンテナンスされていないので、すべての情報が正しいとは限らない。				
指摘内容	レビューア	山田花子	レビュー日	2015/AA/DD	
	構造、処理を把握する調査なので、付録 3 の調査実施事例（探索ルートと成果物）の探索ルートをたどること。調査している関数内に新たな関数が現れた時に、その関数の中に入らないこと。調査範囲が狭まり、広い範囲で構造、処理を把握できなくなる。				
熟練者が事前調査テーマをレビューし記載					

■ 事前調査実施

実施結果	実施者	山田太郎	実施日	2015/AA/EE	別紙	(有) / 無し
	10H を使い、YYY モジュール調査書を作成。YYY モジュールの main 関数から 300 関数を記載した構造図を作成。また処理が不明な関数について、5 つ PAD を作成。この結果、XXX 機能追加ではデータ受信処理はすべて func1() に集約されている。またデータ送信処理は func2() に集約されている。					
指摘内容	レビューア	山田花子	レビュー日	2015/AA/FF		
	データ受信処理 func1() とデータ送信処理 func2() の間に何か処理をする関数はないのか？また新たに追加されたパラメータによって、内部の動作が変わるところはないのか？					
熟練者が事前調査の結果をレビューし記載						

付録 3： 調査実施事例（探索ルートと成果物）

1. 調査結果の残し方

ソースコードの処理を調査する場合、理解してから文書に残すのではなく、関数単位でソースコードを読み、読むという行為と同時に構造図、また必要に応じてPADに変換する。このとき、ソースコードを読んで理解するのではなく、単純に構造図やPADに変換するだけにする。変換していく過程である程度理解が進むので、あとは一段落したところで、改めて構造図やPADを見て理解を深める。このように調査することによって、今後の派生開発に使える資料が残るようになる。

2. 調査事例

以下に、関数 func1() と、そこから呼び出される下位関数を調べる際の例を示す。また、図 5-1 は、その際の探索ルートを表しており、さらに図 5-2 は、その過程で構造図が書かれていく様子を示している。

- func1() を図 1 の①の探索ルートでソースコードを読みながら図 5-2 の①の部分の構造図を記載する。
- func2() も func1() と同じようにしてソースコードを読み、図 5-2 の②の部分の構造図を記載する。
- func3() を調べることで図 5-2 の③の部分の構造図を記載する。その際に func3() の処理が複雑な場合、PADを同時に記載する。PADを記載した時は構造図の func3() の右上に●を残しておく。そうすることで、後でこの構造図を見たときに、func3() のPADがあることがわかる。
- 引き続き func4() を調査することで図 5-2 の④の構造図が記載される。さらに、func5() を調査したとき、そこから呼び出されている関数がない場合は、構造図で func5() の下に“グランドマーク”をつける。グランドマークはその下に関数がないことを示している。これによって構造図にグランドマークがない場合は、まだその関数を調べていないことがわかる。
- 同じように func6()、func7()、func8()、func9() を調べていく。

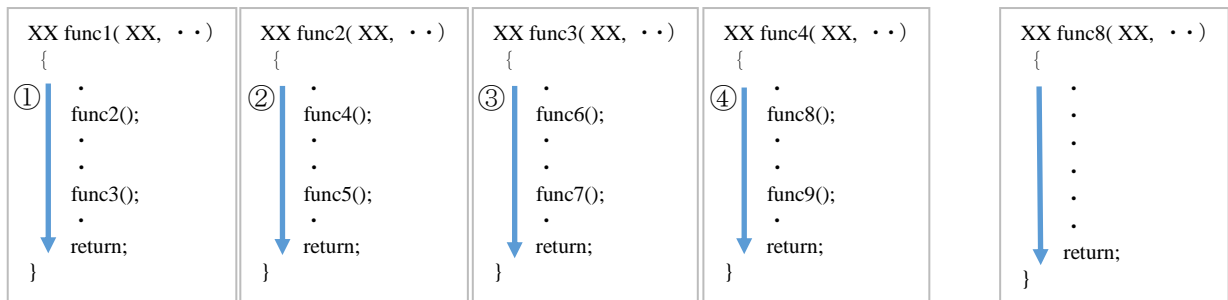


図 5-1. 関数の探索ルート

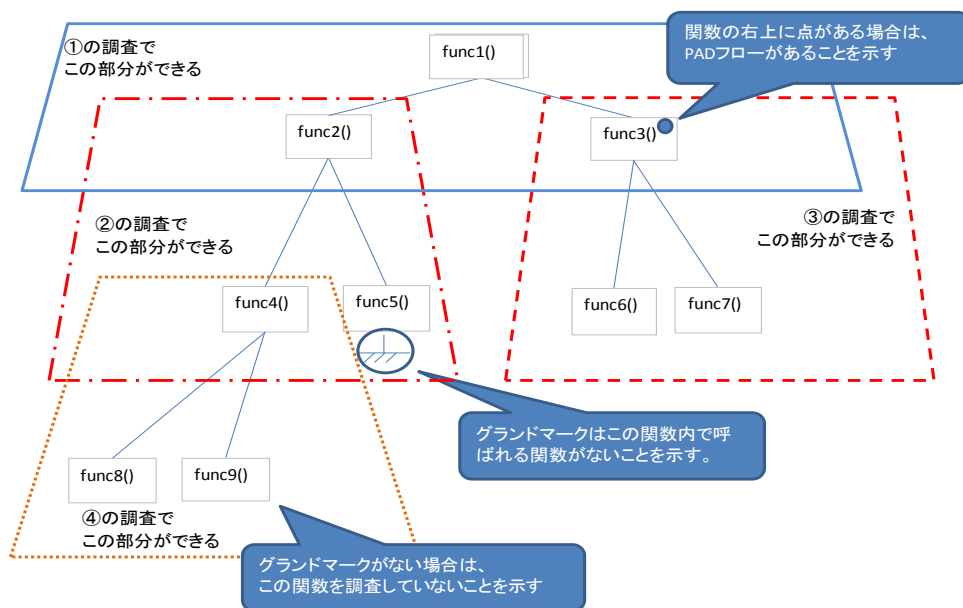


図 5-2. 調査で残される func1() の構造図

3. 時間がかかる調査事例

ソースコードを調査する際に図 5-3 のような探索ルートをとらないようにする。この探索ルートは、ソースコードを読んでいるときに、関数が現れた場合、すぐにその関数に飛んで次の関数を読むことを示す。

このような探索ルートをとって、ソースコードを読んでいくと、次々に関数を見ていくことになり、調査に時間がかかる。また、構図や PAD は残らないので、以降の派生開発で使える資料が残らない。そのため、何度もソースコードを読むことになり、さらに調査に時間がかかる。

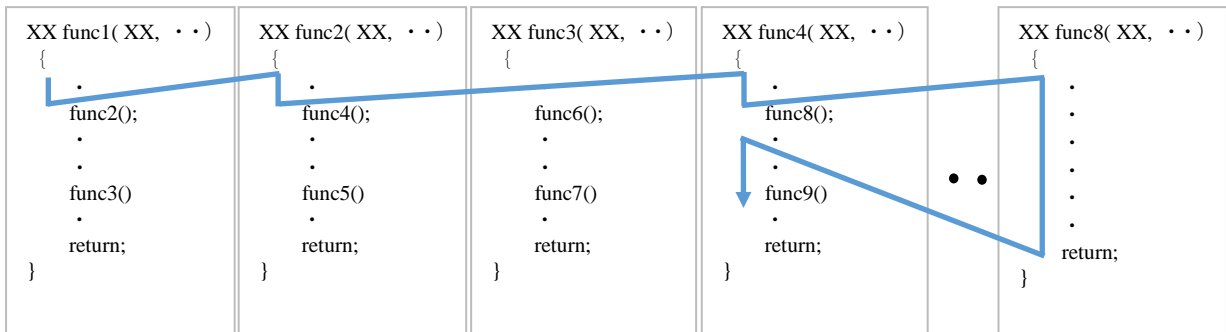


図 5-3. 調査に時間がかかる探索ルート

4. PAD

PAD (Problem Analysis Diagram) とは、日立製作所の二村良彦氏が考案した構造化プログラミングに適した技法である。詳細は情報処理学会論文誌「PAD (Problem Analysis Diagram) によるプログラムの設計及び作成」を参考にすること。以下に PAD のサンプルを示す。

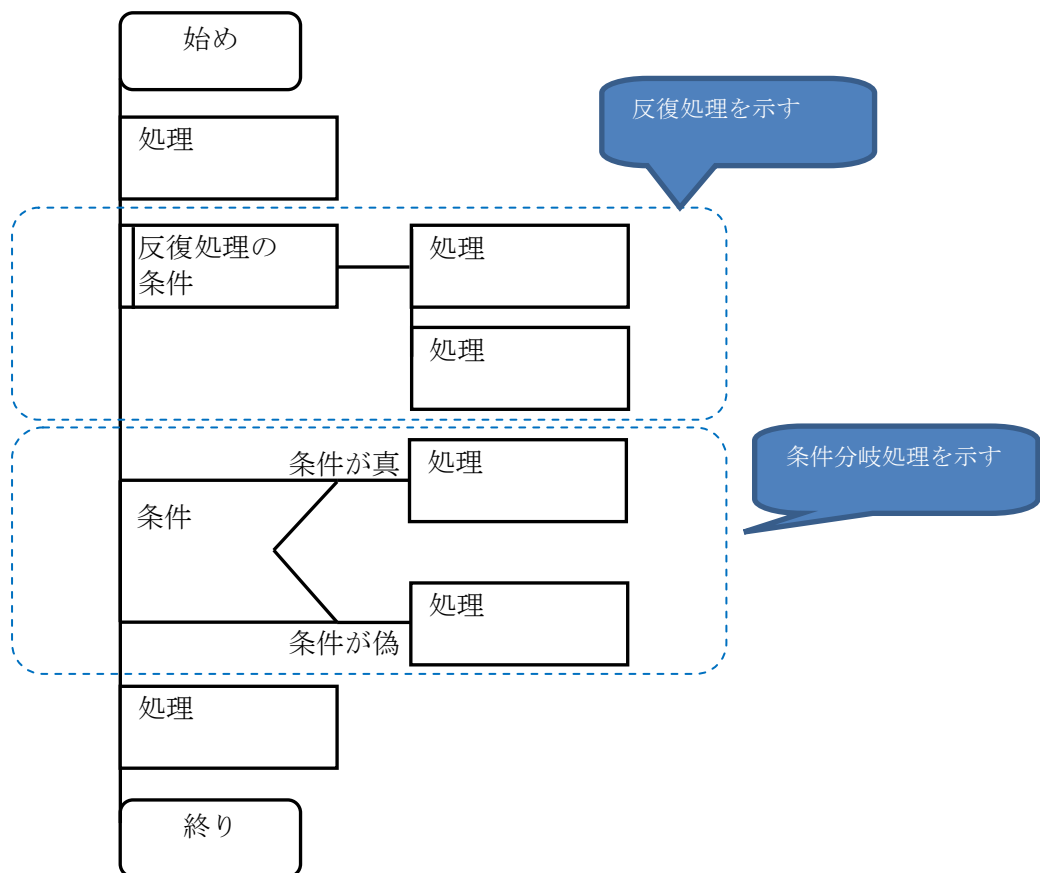


図 5-4. PAD のサンプル