

ソフトウェア要求定義の品質向上技術

石井俊直*
細谷泰夫**
石原 鑑*

On Development and Validation of Software Requirement Specifications

Toshinao Ishii, Yasuo Hosotani, Akira Ishihara

要 旨

近年、ソフトウェアはあらゆるシステムで使われるようになった。そのためソフトウェアへの要求は多様化し、要求の規模も大きくなっている。ハードウェアの制御を中心とした組み込み分野のソフトウェアでも、この傾向は見られている。このため、ハードウェアが持つ機能の観点からソフトウェアへの要求を仕様化する従来のアプローチだけでは、システムへの要求を十分に把握することが難しい場合が多くなりつつある。本稿では、こうした問題に対応する一つの取組みとして、組み込みシステムの分野でソフトウェアの要求定義にシステムを使って行う業務の観点を積極的に取り入れた要求開発の取組みについて述べる。

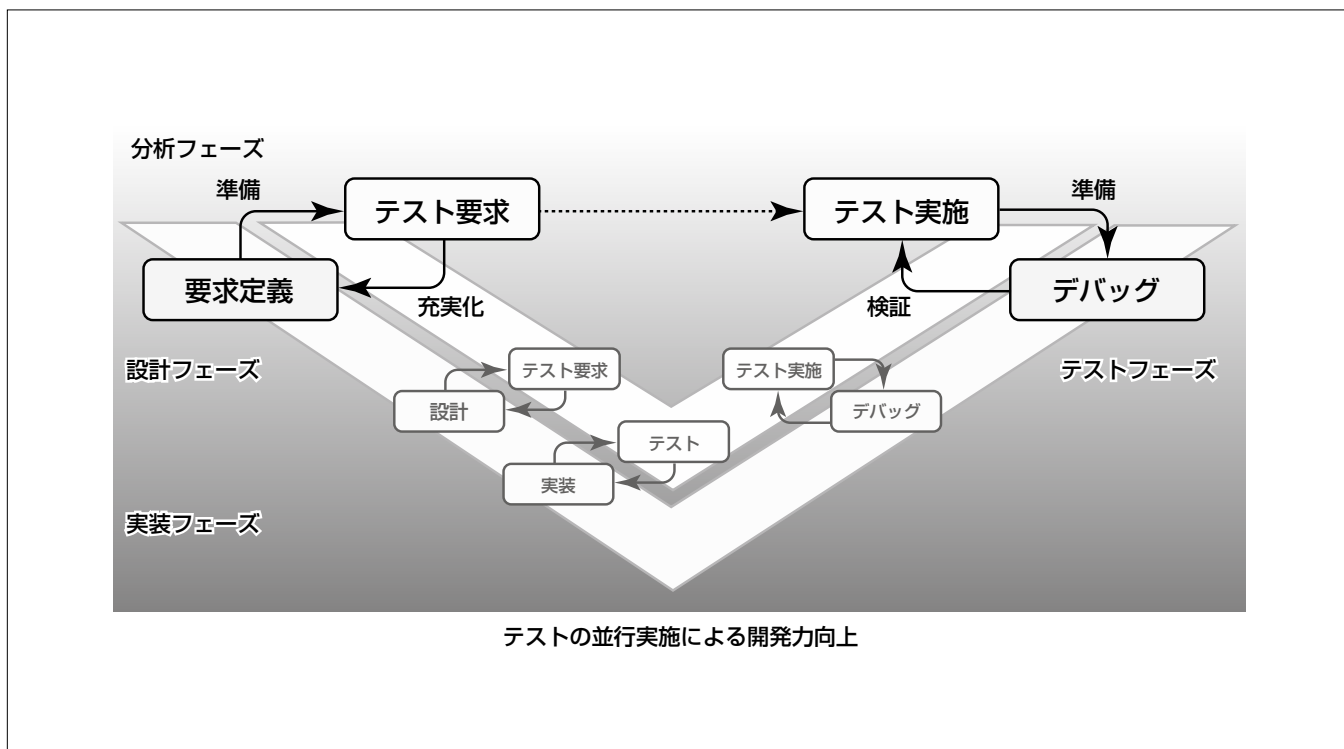
この業務観点からソフトウェアの要求定義を導く第一歩は、一般的なオブジェクト指向ソフトウェア設計(OOSE)と同様、業務でアクタ(ユーザー)とシステムの間で起こる振る舞いを表すユースケースの分析である。その後、ユースケースを利用しテスト要求を分析することで、テストの準備を前倒し実施するとともに要求定義にそのフィードバ

ックを行うようにした。上流でテスト要求の分析を行うことで、網羅性向上などソフトウェアの要求定義が充実化される。また要求定義とテスト実施の関連を確かなものとし、デバッグと要求定義の直接的関係を明確にするねらいもある。

この要求開発プロセスは新しい取組みである。実際にプロジェクトに適用すると課題も多く見つかるが、要求開発への有効な作用を含め、次に挙げることを含め多くのことが確認された。

- (1) ユースケース分析結果が多面的に利用価値が高い。
- (2) テスト設計のために用意されたガイドラインがソフトウェア要求開発で有用である。
- (3) 直交法⁽¹⁾やCFD(Cause Flow Diagram)⁽²⁾などのテスト設計技法がソフトウェア要求開発でも役に立つ。

本稿では、要求定義を行う分析フェーズにおけるテスト準備とそれによる要求定義充実化の取組みについて述べる。



テスト要求や実施を開発のすべてのフェーズに取り入れるプロセス

実装フェーズをテストで駆動するプロセスが一般的になりつつあるが、ソフトウェア開発のその他のフェーズもテストに関する活動を平行実施することで開発力が向上する。

1. ま え が き

本稿では、社内のある組織を対象に行ったソフトウェア要求仕様作成ガイドラインの改訂と適用の経験をもとに、ユースケースやテスト観点の分析をソフトウェア要求定義フェーズで活用する技術について述べる。

ソフトウェアは、ビジネス課題の計算機システムによるソリューションととらえることができる。すなわち、ある計算機システムがユーザーや他の計算機システムと情報のやりとりをすることでビジネス課題が解決されるが、その枠組みを定義し、また情報のやりとりを実現するのがソフトウェアであると考えているのである。例えば、携帯電話機は最先端の計算機システムだが、その中の通話プログラムは、遠くにいる相手と会話をするという課題を解決している。計算機を実際に動かすプログラムのみならず、システム実現方式などの設計もソフトウェアである。課題解決に適した設計は、具体的な計算機システムより本質的なものであるから、設計をソフトウェアから除外する理由はない。設計とプログラムは、解決のレベルが異なっているだけである。ソフトウェアは一般的にこのようなものなので、その開発では様々なレベルで課題とシステム、そしてその境界を定義することが必要になる。

ソフトウェア開発を分析フェーズ、設計フェーズ、実装フェーズ、テストフェーズに分けると、本稿の対象は主に分析フェーズで、特にソフトウェアの要求定義を話題とする。したがって、課題・境界・システムというソフトウェアモデル化の枠組みで、ソフトウェアの要求定義をどのように進めるべきかが、またより具体的に、要求定義にユースケース(UC)やテストの観点がどのような役割を担うかが本稿全体で議論する課題である。

ソフトウェア要求定義プロセスを定めるには、そもそも課せられている制約とプロセス選択の基準とを分析しておくことが必要である。ここでは、ソフトウェア要求定義フェーズへの主な制約として、次の点を考慮した。

- (1) ソフトウェア要求開発の上流工程では、モデル化の視点がビジネス課題寄りになり計算機システムは高いレベルで抽象化されていること
- (2) このたびソフトウェア要求開発プロセス改革を適用した組織では従来、機能を中心とした開発プロセスが主流であったこと

またプロセス選択の基準としては次の点を考慮した。

- ①ソフトウェア機能の定義に結びつきやすいこと
- ②品質の定義が容易になること
- ③ソフトウェア要求定義の妥当性が確認しやすいこと

①と②は制約(2)と関係している。③は、ソフトウェア要求仕様レベルの不具合の流出防止やテスト工程の効率化などと関連する重要な基準である。

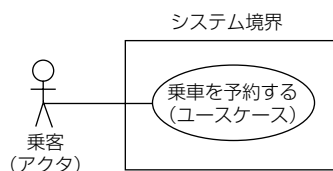
2. ユースケース

一課題領域から見たソフトウェア要求のモデル

業務観点からのソフトウェア要求定義の第一歩は、一般的なオブジェクト指向ソフトウェア設計(OOSE)⁽³⁾と同様、業務でアクタ(ユーザー)とシステムの間で起こる振る舞いを表すUCの分析を採用した。UCは図1に示すようなもので業務が達成されるまでの振る舞いを表したものである。振る舞いと言っても、ソフトウェア要求レベルのUCでは、ユーザーの作業というよりは、起こるべきこと、言い換えるとアクタとシステムの意図を物語風に簡潔に記述したものである。

このUCによって記述した開発対象のモデルには、課題・境界・システムが形式的にはっきりと表れる。アクタが一つのUCの物語で達成することが、すなわち課題に相当する。システムはこの段階では中身がほとんど定義されない一つの箱でしかない。そして境界は、物語におけるアクタとシステムの間でやりとりされる情報という形で明らかにされる。

このようなソフトウェアUCは、ビジネス課題領域からシステムを分析した結果であり、図1上に示した図や図1下の物語風の文書で記述される。この文書もある形式に則(のっと)って記述されたものであって、自由奔放に記述されたものではない。こうした形式化はソフトウェア開発のあらゆるフェーズで重要であり、ツールによるサポートを可能にするのに必要な要件となるが、本稿ではこの問題にはこれ以上触れない。さて、この段階でシステム内部についてはほとんど定義が行われないが、システムが全体としてビジネス課題にどのように作用すべきかが定義される。そして、3章以降で詳しく述べるが、この物語に表れた名詞や動詞がデータや機能としてシステムの内部を形作っていくことになる。



ゴール	乗車を予約する
主アクタ	乗客
	1. 乗客は乗車システムにログインする。
	2. システムは、乗車希望条件入力画面を表示する。
	3. 乗客は、希望条件を入力する。
	4. システムは、希望条件に合う運行予定を表示する。
	5. 乗客は、表示された運行予定から乗車予約希望を選択する。
	6. システムは、乗車予約を行う。
拡張シナリオ	7. システムは、乗車予約情報を表示する。
	4a. 乗車希望に見合う運行予定がない場合…

図1. ユースケースの記述例

3. 機能の定義

一般的なオブジェクト指向設計プロセス⁽³⁾では、UC分析結果をもとにして、より形式的なモデルへの詳細化が行われる。こうしてできたモデルは分析モデルと呼ばれる。分析モデルを導く代表的な手法にバウンダリ分析がある。しかしながら、我々はこの方法を採用しなかった。代わりに、UC分析結果をもとにソフトウェアが実現すべき機能一覧を作成することとした(図2)。ここで機能とは、およそUCの記述に現れた動詞に対応するものである。例えば、UCに“システムはアクタに〇〇を表示する。”という記述があれば“〇〇表示機能”が要求機能の一つとなるという具合である。

このようにしたのは、この定義スタイルが、プロセス変革を適用した組織が従来多く採用してきたスタイルに近いことが理由の一つである。機能項目化のあと、“□□機能”がすでに確立されたコンポーネントとして再利用可能なものであれば、その再利用を行うことができる。また“機能”というスタイルに経験豊富なエンジニアはよりスムーズにその経験を生かすこともできる。一方でいったん機能として定義したところで、さらにOOSEの手法を適用していくことが妨げられるものでもない。

また、機能として分解することで4章で述べるテスト観点に基づく分析を加えることが容易になる。このテスト観点を使った分析は、振る舞いレベルにとどまったままで要求仕様を充実化することに効果が大きい。

UCから機能を抽出すると、機能は自然にUCに束ねられて項目化される。4章で、テストへの要求について述べる中でこの性質を利用する。このほかにも、UCに束ねられることで、個々の機能が利用されるコンテキストが具体的になり、ソフトウェアへの品質要求がより明確になるといったメリットもある。

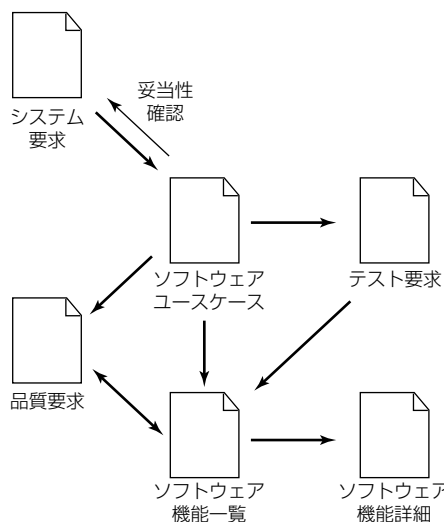


図2. UCを中心としたソフトウェア要求仕様の詳細化

4. テスト要求の分析とフィードバック

ソフトウェア要求定義のフェーズでテストへの要求分析を前倒しに実施する具体的メリットには、次の二つがある。

- ①プログラムのテスト準備の早期開始
 - ②ソフトウェア要求定義の妥当性向上、要求漏れの低減
- 開発プロセスのVモデル^(注1)に加え、早期のテスト準備などの要素を明記したWモデル^(注2)の提案者スピルナーは、Wモデルの本質が“テストと開発の同時平行性”であると述べている⁽⁴⁾。言い換えると、開発のどのフェーズであっても、フェーズによって役割こそ変わるが、テストと開発はいつも同時並行的に行われるのがWモデルに則った開発プロセスである。要求定義の段階でWモデルのプロセスを具体化するため、テスト観点分析を導入した。テスト観点とは、ソフトウェアやシステムを実行する際の環境、結果、実行の部分などのコンテキストから、テスト設計で利用するために特定されたものと我々は定義している。ここで例として、仮想的なモバイル端末のテスト観点分析を行ってみよう。端末には初期化という実行の部分があるだろう。その過程で接続しているネットワークの種類や容量にバリエーションがあると考えられる。また、モバイル端末のバッテリーの状態や、接続されている周辺機器も同様にバリエーションがある。このような分析を行うと、初期化、ネットワークタイプ、ネットワーク速度、接続周辺機器などのテストすべき観点が明確化されてくる。

実際にテスト観点分析を行う場合には、テスト専門家が作成していたテスト設計のためのガイドラインに、テスト観点を特定するためのヒントが散りばめられ有用であった。テスト観点の分析は、UCのシナリオに沿って検討を進める形で進めた。一つ一つの機能はUCシナリオに比べると断片的であるため、コンテキストがイメージしにくい。UCシナリオを活用することで、テスト観点の母集団としてより網羅性の高いものを見いだすことができた。

テスト観点を分析した結果は、マインドマップ^(注3)の形でまとめた。さらに、テストすべき処理フローやそのための条件について分析を進めると、要求仕様の妥当性確認テストにつながる。テスト観点分析結果は、このようにテストへの準備としての役割を持つが、さらに分析結果をフィードバックすることで、ソフトウェア要求の充実化がなされる。ソフトウェア要求開発の立場からは、このことの方が重要と言える。はじめにUCをもとに行うソフトウェア要求分析は、ソフトウェアの振る舞いのうち、典型的な業

(注1) V字の底に実装を置き、左側に要求定義と設計を、右側にテストを記して表すモデル。一方向に開発が進むウォーターフォールモデルにおける対応関係を示している。
 (注2) 本稿扉ページの図で示したような、Vモデルに平行な二番目のVを持つモデル。
 (注3) マインドマップは、Buzan Organization Ltd. の登録商標である。

務シナリオが中心とならざるを得ない。これに対し、ソフトウェア要求開発では、それ以外を含めた様々な例外的振る舞いについての要求仕様化も行われるべきである。テスト観点分析は、業務観点の分析で漏れがちな例外的状況への要求を特定するために効果があった。

5. プロジェクトへの適用

最後に、実際の開発プロジェクトへの適用について簡単に触れる。改訂したソフトウェア要求仕様書作成ガイドラインは、現実に行われていることを整理したものというよりは、一つのあるべき姿をもとに作成したものという色彩が強い。そこでまず、規模や対象がこの新しい考え方に合っているプロジェクトを選択し、プロセスエンジニアが積極的にプロジェクトとかかわりながら、全面的に新しいプロセスでプロジェクトのソフトウェア要求開発を進めるといった試みを行った。このほかに組織全体への説明会も実施したが、プロセスエンジニアリング側のプロジェクトへの直接的にかかわりは、新しい方法の普及に不可欠であるように思われた。

テスト観点分析を実施した結果、UCごとにシナリオ数、テストケース数、処理フロー数を比較することでUC単位で要求仕様書の充実度を推定することができた。テスト観点分析前と後を定量的に比べると、UCシナリオ数と比べおよそ1.7倍のテストケース、シナリオ数で3%程度の不可欠なUCシナリオを新たに見いだすことができた。また、テスト観点分析からおよそ1,000通りのリソース競合関係を特定し、結果的にはこの中のおよそ3%が開発ソフトウェアに特有の競合処理定義が必要であったが、それらすべてをテスト観点分析をもとに定義することができた。

このほかの取組みとして、新しいガイドラインの部分適用も試みた。これは、徐々に浸透させていくという戦略であるが、テスト観点分析は取組み自体が新しいものである

ため、この領域がいわば“更地”となっていることが多く、部分導入の差し込み口として適していた。新しいプロセスを組織レベルで導入するには、通常段階的な移行プロセスが必要となる。ところが、今までやってきたことを変えることはなかなか難しいもので、移行が進まないことはよく見られる。今回の取組みでは、更地の開拓の方がすでにいろいろ積み上げのあるところを変えることよりもスムーズに進む傾向が見られた。このことから、有用な新しい技術をしてこにした移行は、一つのパターンになるだろう。

6. むすび

UCによる分析を中心としたソフトウェア要求仕様開発の新しい取組みについて述べた。この取組みで採用した開発プロセスは、ソフトウェアへの要求を洩(も)れなく見つけることに重点を置き、そのため様々なテクニックを使いソフトウェアの振る舞いについて追求するものである。

要求を的確に獲得することは、顧客に満足いただけるシステムを提供するための基本である。ここで述べたプロセスを普及・発展し、顧客により高いレベルの満足を提供できるよう、今後も活動を進めていきたいと考えている。

参考文献

- (1) 立林和夫：入門タグチメソッド，日科技連出版社（2004）
- (2) 松本正雄，ほか：ソフトウェア開発検証技法，電子情報通信学会（1997）
- (3) イヴァー・ヤコブソン，ほか：UMLによる統一ソフトウェア開発プロセス—オブジェクト指向開発方法論（Object oriented selection），翔泳社（2000）
- (4) Spillner, . A, et al. : Software Testing Practice : Test Management, Rocky Nock Inc. (2007)