特集論文

鉄道車両情報管理装置における ソフトウェア開発プロセス改善

伊丹事業所　技術第1部　車両情報システム第2課　藤原　祐貴、崎山　智代
システム開発課　井筒　啓満

1. まえがき

当所では、鉄道車両搭載機器の情報を統合制御し、機器監視、制御、乗務員支援を行う、車両情報管理装置（以下、TCMS（Train Control and Management System）のソフトウェア開発を行っている。電鉄各社では、より快適な車内つくりや安全性の向上、メンテナンスの効率化を常に図っており新車投入後の車両改造が多く、それにともないソフトウェア改造の機会が多い。そのため当所のソフトウェア開発は派生開発が中心で行われている。

近年TCMSは、高度な機能を求められるようになり、ソフトウェアは次第に複雑となってきた。また電鉄各社からは納期の短縮要求や、TCMSが障害に陥った場合、電車が止まるなど社会的影響が大きいことから従来以上の高い品質を強く求められるようになってきた。

一方、開発現場では長年の改変積み重ねによりプログラムは複雑化･肥大化する中、従来の開発手法から脱却できず工数だけが増大していた。客先のニーズに応えるには大幅な品質向上及び効率化が必要であり、そのためには抜本的品質改善が必要であった。そこで当所では、2015年下期より派生開発特有の「短納期」や「部分理解」といった問題に合理的に対応する方法として提供された開発アプローチであるXDDP（eXtreme Derivative Development Process）を導入し試行を行った。現在ではTCMSの開発スタイルに沿うよう、伊事版XDDPを確立させ品質向上に大きな成果を上げることができた。

本稿では、従来の問題点を伊事版XDDPの構築によってどのように改善できたか、また効率化に向けての改善活動について紹介する。

2. 従来のソフトウェア開発の問題点

2.1　品質データからみた問題

従来は、ソフトウェア品質を向上させるべく様々な対策を講じたためソフトウェア開発工数は肥大化する一方であった。しかし多数の対策を講じているにも関わらず、障害の件数や手戻りの工数は減少傾向にあるものの撲滅には至っていない。また、障害はソフトウェア設計フェーズで作り込まれているものが多く、そしてその要因は図1に示すように「流用元ソフトウェア仕様の理解不足」や「流用時、影響範囲の検討不足」が大半を占め、従来の改善策では効果は少なかった。

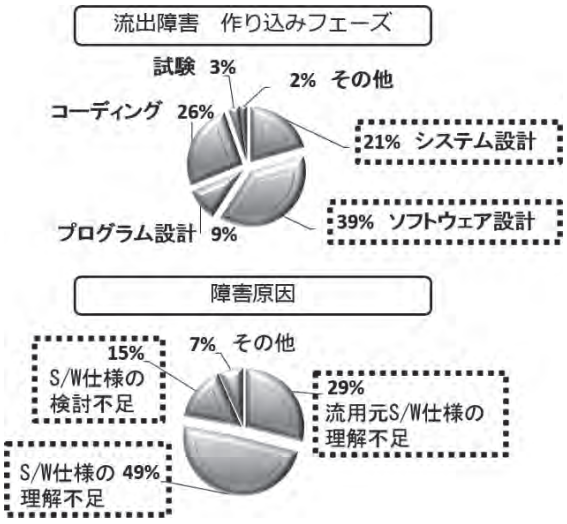

図1. 障害要因内訳

2.2　インプット資料の問題

ソフトウェア設計へのインプット資料は、1システムにつき多く分冊されており、図2に示すようにそれぞれの仕様書と紐付けられているが、多くの仕様書から関連を整理することに時間を要していた。

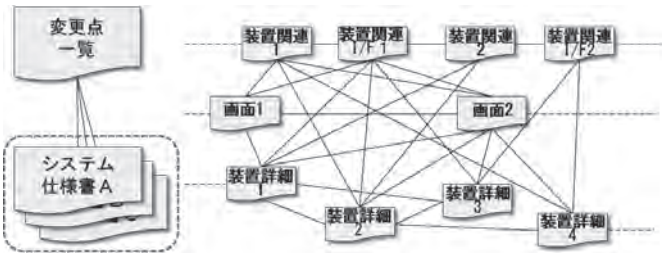

図2. システム仕様書体系

また、図3に示すように変更箇所がまとまっておらず変更箇所の確認が困難であることや、システム仕様書には要求及び変更理由の記載がなく結果のみが記載されるため、なぜ変更するのかが不明であり、システム設計者の意図と齟齬が発生していた。

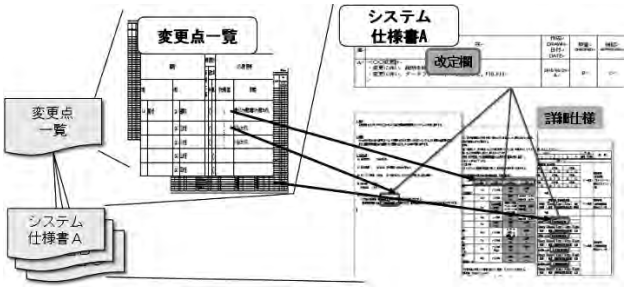

図3. システム仕様書の変更箇所

2.3 受注フェーズ内の問題

(1) 既存ソフトウェアの解析が困難

TCMSの開発当初は、現在よりも開発規模が小規模であったため、ソフトウェア設計後、設計書作成と同時にソースコードへ追加や改造を行う場合があった。そのためソフトウェアドキュメントの内容が局所的であったり、漏れが発生しやすく、開発が複雑化・大規模化していく中で、図4に示すように開発者はソフトウェア全体像を理解できないまま変更箇所のみを仕様書化することとなり、変更漏れや変更方法の誤りが発生していた。

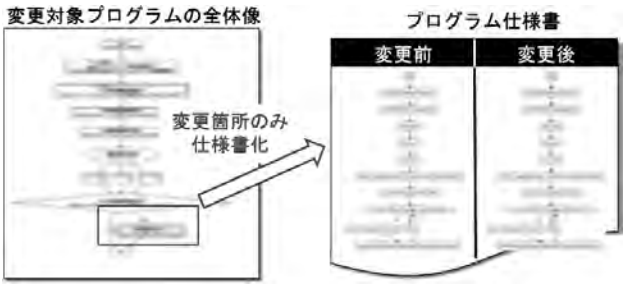

図4. 従来のプログラム仕様書

(2) ソフトウェア設計書の検証が不十分

ソースコードとの対応をとりながらソフトウェア設計書を作成しているが、変更点が多く管理が不十分であった。また図5に示すようにデザインレビューの実施タイミングや回数も不適切であった。

図5. 従来のソフトウェア開発プロセス

(3) 非効率なプログラム変更作業

同一の改造工事内で、仕様書を受領するつど局所的にプログラムを変更していた。図6に示すように同じ担当者が同じプログラムファイルを何度も開けてプログラムを変更することがありムダや誤りが多く発生していた。

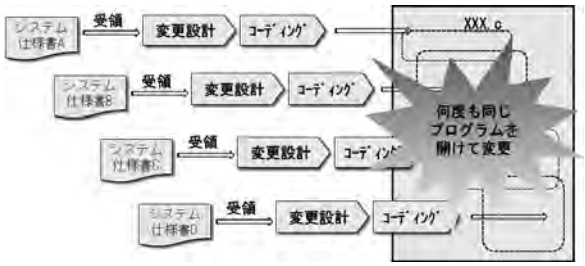

図6. 従来のプログラム変更プロセス

3. XDDPの導入

3.1 XDDPとは

XDDPとは、「変更」に特化した「派生開発」のためのプロセスモデルで、3つの成果物（図7、以下XDDP3点セット）となる変更要求仕様書（以下USDM（Universal Specfic Description Manner）(注1)）、トレーサビリティマトリクス（以下TM）、変更設計書により、モレ・ミス・ムダを防止でき、そして3つの成果物作成後に一斉にコーディングを行うことで効率化が図れるのが特徴である。

そしてXDDPを導入することによって期待される効果は以下のとおりである。

① USDMでは要求と仕様を形式化することにより、要求仕様の抜けや漏れ、矛盾などの問題点を見つけやすくする。

② TMではUSDMで定義した仕様に対するコードをマトリクス表で対応付けることにより、変更仕様に対する影響範囲を把握することができ、変更箇所の抜けや漏れを防ぐ。

③ 変更設計書では変更箇所に対してどのような変更をするのかを設計し、それをもとに有識者と綿密にレ

（注１） 要求文から要求を分割・階層化し、要求仕様・理由・説明をセットにして、表形式で仕様書を記述・管理すること。

ビューすることによりミスを防ぐ。

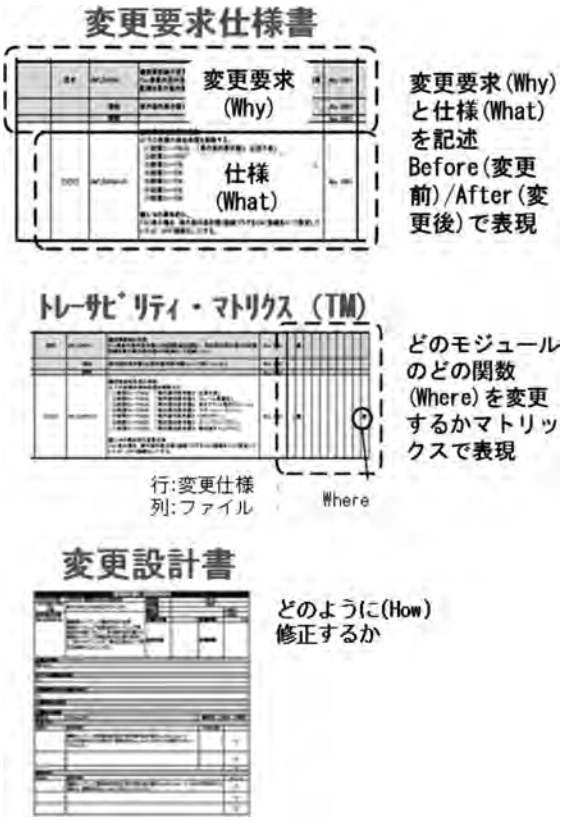

図7. XDDP3点セット

3.2 XDDPの活用

当所では、2.2項で述べたインプット資料の問題についてはUSDMを活用することにより、変更箇所をモレなく抽出でき、変更理由を記載することでシステム設計者の意図と齟齬なく要求と仕様を明確にできると判断した。また2.3項で述べた受注フェーズの問題については、USDMとTMを活用することにより変更する全体像が把握でき影響範囲を見渡すことができる。そして変更設計書で変更内容を具体化することによりレビューやテストがよりきめ細かく実施できると判断しXDDPの活用を進めた。

一般のXDDPは、要求定義から組合試験までをスコープとしているが、「伊事版XDDP」では受注範囲にあるソフトウェア設計から組合せ試験をスコープとし、3.1項で述べたXDDPの特徴をもとに、XDDP3点セットの内のUSDM及びTMをTCMSのソフトウェア開発に最大限活かせるようカスタマイズして、「伊事版XDDP」を構築した。

第一に変更点一覧の全体概要をシステム要求、変更概要の記載内容をなぜ変更が必要なのか、または何を実現したいのかといった理由とし、次にシステム仕様書の内容をシステム仕様(ソフトウェア要求)と捉えるようにした。それを図8に示すように階層化して整理・明確化するようにした。そしてこの要求と仕様をもとにソフトウェア仕様を設計することにした。この手順により散らばったシステム仕様をUSDMへの記載でシステム要求からソフトウェア仕様までを一環して把握することができ、レビューでの変更モレや影響範囲の検討モレへの気づきを強化することが可能となった。

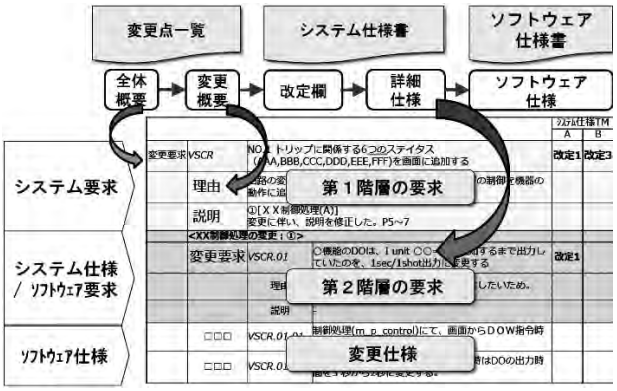

図8. 伊事版XDDPのUSDM

TMでは図8のUSDMに図9で示すようにテスト観点を追加することにより、観点に応じた組合せ試験設計を行うことができ、組合せ試験の網羅性を高めた。

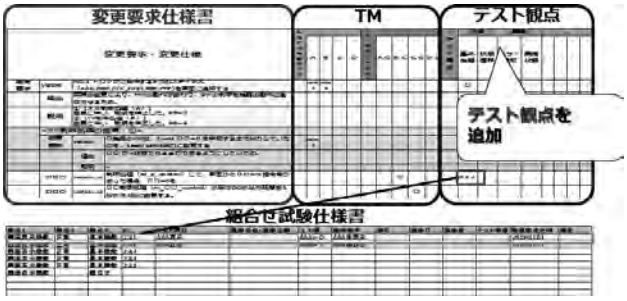

図9. 伊事版XDDPのTM

3.3 伊事版XDDPの導入結果

伊事版XDDPを構築し運用した結果、組合せ試験以降の1案件あたりの障害件数が、同規模の案件と比較すると大幅に削減することができた。また、障害除去状況についてみても、図10に示すように従来では検出できず試験フェーズまで持ち込んで刈り取っていた障害が、ソフトウェア設計で刈り取ることができるようになった。これはソフトウェア設計段階でシステム仕様の不明点が明確にできてきたことから、システム設計者への問合せや、システム仕様の理解不足、思い込み、勘違いによる手戻りが減少したことによるものである。

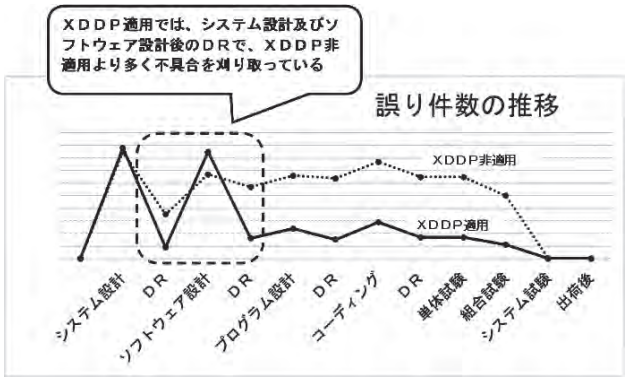

図10. 4岳グラフ比較

その結果、2017年度完了の適用工事40件（適用率約50%）についてFL化率(注2)は平均87%の結果を得ており、目標値の80%を大幅にクリアすることができた。図11はその適用工事（一部抜粋）のデータである。

工事名	SW設計～組試まで 見積り時間に対する実績時間の割合(%) ＊実績時間/見積り時間	FL化率(%)
A	110.5	83
B	72.7	93
C	57.1	67
D	72.9	90
E	60.3	90
F	164.4	85
G	96.9	90
H	86.4	85
I	70.0	100
J	70.9	100
K	118.0	100
L	109.0	100
M	113.5	96
N	99.9	86
O	99.1	77
P	78.0	88
Q	88.0	100
R	80.0	90

図11. 伊事版XDDPの適用評価一例

また、下流フェーズでの手戻り時間が減り、ソフトウェア開発全体の時間が短縮されたことで、変更内容の再確認や組合せ及びシステム試験などに時間を費やすことができたため、出荷後障害の削減につながったと言える。

ただし、2017年度の適用率が約50%であること、一部案件では目標値が未達成であったため、今後適用拡大するとともに精度を高めて改善に取組んでいく。

4. USDMを利用した大規模案件への展開

4.1 USDM形式のソフトウェア設計書の確立

3項で述べたように、当所では「伊事版XDDP」を構築し、派生開発を対象として「変更点」に着目しソフトウェア設計品質を高めてきた。しかし、2018年に客先の主力路線への新車投入にあわせて、大規模なTCMSの新規案件が予定されていたため、当所ではこの新規案件についても「伊事版XDDP」を適用して従来型のソフトウェア開発プロセス（ソースコードベースの作り込み）から脱却し、ドキュメントベースの品質作り込みを行うべく、一からUSDMを利用したソフトウェア設計書を確立してソフトウェア開発を進めた。進め方は図12で示すように各STEPに分けてSTEP毎に作業を行った。

STEP1：流用元ソースコードから実装仕様を表現する。
STEP2：実装仕様から機能仕様に変換する。
STEP3：要求と仕様を分離してUSDMで記述し、ソースコードのトレースを取る。
STEP4：システム仕様書から要求を抽出し、異なる要求をUSDMに反映する。
STEP5：抽出した要求をもとに仕様を変更し、ソースコードの変更要否を可視化する。

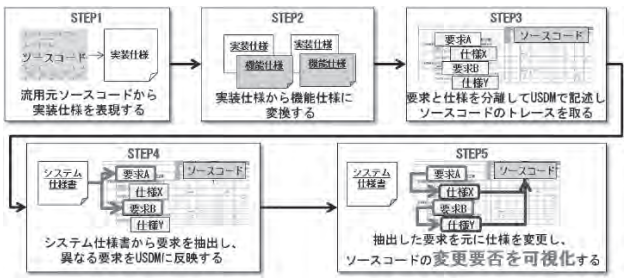

図12. 新規案件への適用手順

次にシーケンス図を作成し、機能のインプット・アウトプットを明確にするとともに、図13に示すようにUSDMに記載した要求仕様と紐付けを行うことで、機能の動作の中のどの部分が作成したシーケンス図に該当するかをわかりやすく表現した。

（注２） フロントローディング化率の略。システム開発におけるフロントローディングを定量的に示すために用いている指標
FL化率＝設計･製作工程の摘出誤り件数
÷全工程の摘出誤り件数）

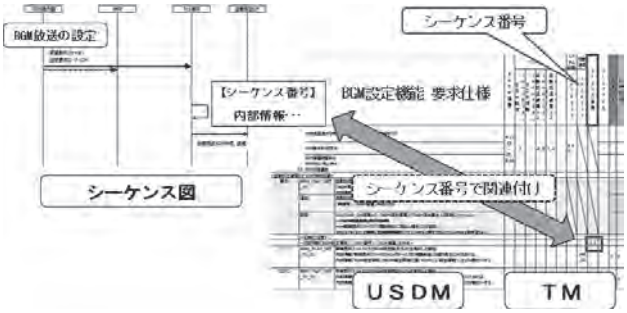

図13. ソフトウェア設計書

4.2 大規模案件への適用結果

今回の新規案件と同規模の過去案件を従来型で行った場合のFL化率を比較した結果、64%から86%（図14）へと大幅に改善することができた。これは要求仕様を可視化し、ソフトウェア設計を行い、ドキュメントベースでレビューを行ったことによる結果である。また、今回の新規案件では現地流出障害は、大規模案件にもかかわらず軽微なもの1件のみであり、品質を大幅に向上することができた。

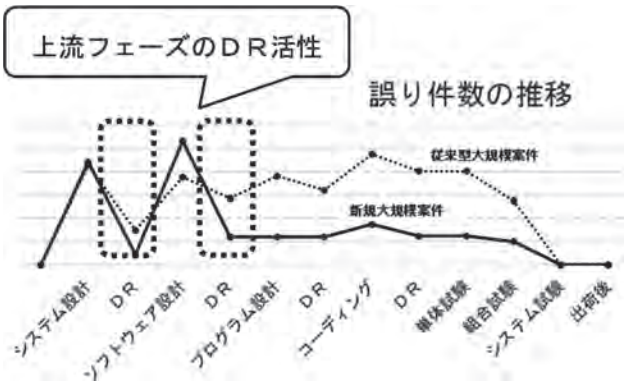

図14. 4岳グラフ比較②

5. さらなる改善に向けて

5.1 ソフトウェアプロダクトライン開発

ソフトウェアプロダクトラインとは、対象とする製品群の共通部分と差異を把握し、その共通部は再利用し、差異を効率的に開発するための手法である。これは、個々の製品開発での最適化ではなく、系列製品の開発全体での最適化を狙った開発パラダイムで個々の技術の枠を超えた、総合的な取り組みであり、以下の効果が見込まれる。

① 開発工数・期間の短縮による短期での製品の市場投入

② 製品品質の向上

4項で述べたUSDM形式でソフトウェア設計書を記載することで、同種類の異なった製品において、ソフトウェア仕様を比較し差異を明確にすることができ、以下に示すプロダクトライン開発の道が拓けるようになった。

例えば、電鉄会社では様々な車形の電車を開発しており、それに対応するソフトウェアは全て異なる。しかし、異なる車形でも基本的な考え方は各電鉄会社にあり、共通機能が少なからず存在している。したがって図15に示すように機能毎に車形を横並びにして比較し、共通部と可変部、固定部を切り分け、共通部を資産として再利用を行うことで効率化を図っていく。

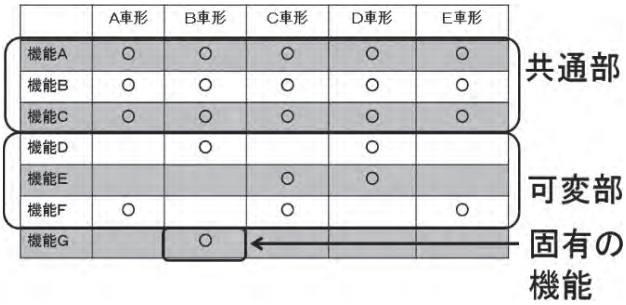

図15. 共通部・可変部の明確化

5.2 ソフトウェアプロダクトラインの手順

新規案件でUSDM形式で要求と仕様を明確にし、ソフトウェア仕様からソースコードまでを管理することができた。これらを利用して図16に示すように各STEPに分けて、STEP毎に作業を行う。

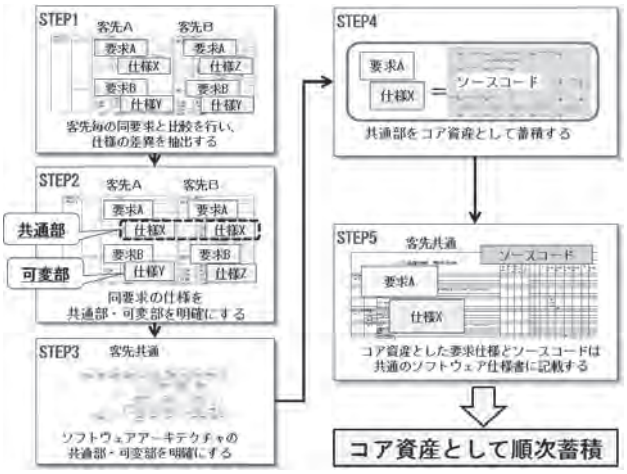

図16. プロダクトライン開発手順

STEP1：車形毎の同要求と比較を行い、仕様の差異を抽出する。

STEP2：同要求の仕様を共通部・可変部を明確にする。

STEP3：ソフトウェアアーキテクチャの共通部・可変部を明確にする。

STEP4：共通部をコア資産として蓄積する。

STEP5：コア資産として要求仕様とソースコードは共通のソフトウェア設計書に記載する。

上記STEP1から5を繰り返し行い、コア資産として蓄積していく。

現在新規案件をもとに図17に示すように他の車形について、継続して試行中である。

図17. 同様機能の車形比較で共通部・可変部の導出

6. むすび

本稿で紹介したXDDPを導入したことで、当所のソフトウェア開発プロセスが、ソースコードベースの作り込みからドキュメントベースの品質作り込みの文化に変貌をとげ、品質優先のプロセスに変わった。これにより品質が大幅に改善された。今後、ドキュメントとソースコードの資産化を行うことで、品質向上に加え効率化にも成果が出るよう進めていく。

さらにソフトウェアの資産化には、システム設計での共通化を図ることが重要と考える。今後システム設計部門とより一層の連携を深め、システム設計からのXDDPの導入とソフトウェアプロダクトライン開発を積極的に提案し実現させる。

執筆者紹介

藤原　祐貴　　フジワラ　ユウキ
2008年入社。主にTCMSのソフトウェア開発に従事。現在伊丹事業所技術第1部車両情報システム第2課。

崎山　智代　　サキヤマ　トモヨ
1998年入社。主にTCMSのソフトウェア開発に従事。現在伊丹事業所技術第1部車両情報システム第2課グループリーダ。

井筒　啓満　　イヅツ　ヒロミツ
1998年入社。主にTCMSのソフトウェア開発に従事。現在伊丹事業所技術第1部システム開発課グループリーダ。