

# 車載組込みソフトウェア開発のコンカレント・エンジニアリング -プロセス変更の柔軟性と製品品質への影響-

日大生産工 ○水上祐治

## 1 はじめに

車載組込みソフトウェアは、自動車用部品に組み込まれるという特性上、出荷後に不具合が発生した場合、製品自体を交換するなど、多大な損失に結びつく可能性が高い。そのため、日本の自動車業界では、高品質の車載ソフト開発が求められている<sup>1)2)</sup>。

組込みシステム開発では、ハードウェア設計とソフトウェア設計が同時並行的に行われる。また、日本の自動車産業では、ハードウェア設計のフロントローディング化が進んでいる。ハードウェア設計がフロントローディングに傾注するのは、開発の後段になるほど、設計変更時に、外注部品や量産設備の変更が難しくなるためである。一方、ソフトウェア開発は、量産設備が不要で設計変更は相対的に容易であると認識されている<sup>3)</sup>。この構図は、ハードウェアを含むシステム全般の設計変更を車載ソフトが吸収する役割を担い、車載組込みソフトウェア(以下、車載ソフト)開発がプロジェクトの末期まで続く原因となっている<sup>4)5)</sup>。そして、この車載ソフト開発の相対的なバックローディング傾向は、車載ソフトの品質低下の一因となっている<sup>4)5)</sup>。

本稿では、ソフト開発のフロントローディング化とその効果である品質向上を目的として、コンカレント型の反復 V モデル Chain-Linked Iterative V-Model (以下、CLIV) を車載ソフト開発へ適用し、ソフトウェアエンジニアリングとプロジェクトマネジメントの両面から考察する。

なお、本稿では、コンカレント・エンジニアリングは、高品質、低コストの製品を短期間で開発するために、製品開発および製造工

程に対して統合された並行処理環境を実現しようとするシステムティックなアプローチとする。このアプローチは、製品ライフサイクルのすべてのフェーズにおいて適用されるよう、開発者に企画段階から考慮させるべく意図するものである<sup>6)7)</sup>。

## 2 本稿が提唱する車載ソフトの開発モデル

日本の車載ソフト開発では、図 1 に示す、一般的な V モデルであるシーケンシャル型の反復 V モデル Traditional Iterative V-Model(以下、TIV)の適用が多い<sup>6)</sup>。しかし、ハードウェア設計のフロントローディング化に伴う、車載ソフト開発の相対的なバックローディング傾向と、その影響による車載ソフトの品質低下が問題になっている。

本稿では、車載ソフト開発のフロントローディング化とその効果である品質向上を目的とし、Kline のチェーンリンクドモデル<sup>8)</sup>と回帰ユニットテストを適用したコンカレント型の反復 V モデル CLIV を提唱する。CLIV でのアクティビティは、仕様分割、仕様実現、仕様統合に大別できる。

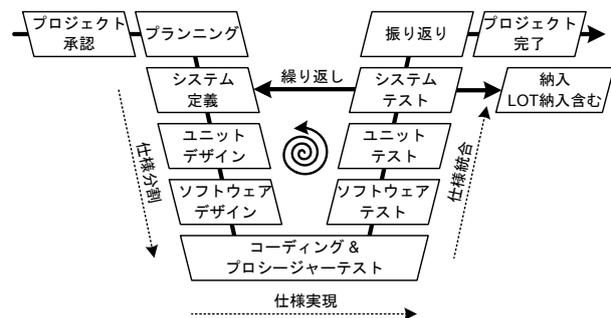


図 1 Traditional Iterative V-Model

A proposed of Concurrent Software Development Process  
in Automotive Embedded Software Development

- Flexibility of the Process Changes and its Impact on Product Quality -

Yuji MIZUKAMI

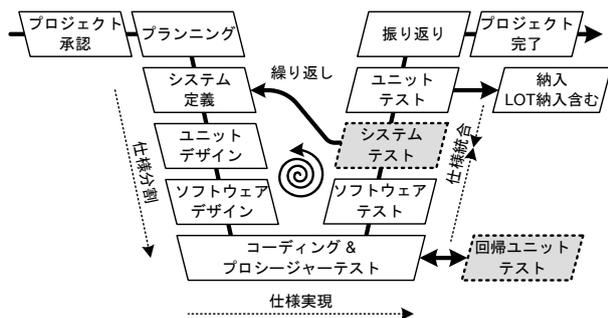


図 2 Chain-Linked Iterative V-Model

### 仕様分割：部品レベル仕様の理解

仕様分割は、車両レベルの仕様を、部品レベルの仕様に分解する工程である<sup>6)</sup>。図 2 に示したとおり、アクティビティは、システム定義、ユニットデザイン、ソフトウェアデザインの 3 種類である。システム定義では、主に自動車メーカーが、車両レベルの仕様を定義する。次に、ユニットデザインで、自動車メーカーと一次サプライヤが協業し、部品レベルの仕様に分解する。そして、ソフトウェアデザインでは、一次サプライヤが、部品レベルの仕様をソフトウェア仕様に分解する。

### 仕様実現：コーディング

仕様実現は、一次サプライヤが、ソフトウェア仕様をもとに、部品の組込みソフトウェアを開発する工程である。図 2 に示したとおり、アクティビティは、コーディング、プロシージャテスト、回帰ユニットテストの 3 種類である。ここでは、コーディング規約、MISRA-C 準拠のツールなどを用いた構文チェックや静的ソースコード解析でソフトウェアの品質を高めることが重要である。

コーディングでは、ソフトデザインを実際のプログラムコードに起こす。次に、プロシージャテストでは、プログラムコードを一行ずつ意図した通りに動作するか確認する。最後に、回帰ユニットテストでは、欠陥修正自体を確認するのではなく、欠陥修正、または、新機能追加時に、別の欠陥が混入し、既存の機能に不具合が生じていないか確認する。

### 仕様統合：車両レベル機能の実現

仕様統合では、一度分割した部品レベルの仕様を車両レベルの仕様に再統合し、開発した機能が、部品レベルと車両レベルの仕様どちらにも合致していることを検証する工程である。図 2 に示すとおり、アクティビティは、ソフトウェアテスト、システムテスト、ユニ

ットテストの 3 種類である。

ソフトウェアテストでは、ソフトウェア・アーキテクチャを理解した上で、意図した通りに動作するか確認する。次に、システムテストでは、車両レベルの仕様検証として、開発した部品を持ち寄り(一時的なコロケーション戦略)、他の部品と組み合わせ、協調動作に着目しながら様々な入力に対して仕様書通りの結果が得られるか確認する。このテストは、車両レベルの仕様を確認する重要な工程である。最後に、ユニットテストでは、部品の入力値と出力値に着目し、様々な入力に対して仕様通りの結果が得られるか確認する。

### 3 本稿が提唱する CLIV の特性

CLIV は、Kline のチェーンリンクドモデル [ ] をソフトウェア開発プロセスに適用したものである。Kline のチェーンリンクドモデルは、技術イノベーションの出発点は「市場の洞察・発見(Market finding)」であるとするデマンド・プル型モデルであり、このモデルのイノベーション・プロセスには、市場の情報を吸い上げるための、多くのフィードバックループが存在する<sup>8)</sup>。このモデルを車載ソフト開発にあてはめた場合「市場」に相当するのは「システムテスト」である。このテストの目的は、自動車メーカーが示した車両レベル機能の検証である。すなわち、CLIV は、車両レベル機能を早期に評価し、検証と改良を積極的に行い、製品および部品の品質と機能性を高め、その結果、これらの市場競争性を高める開発モデルである。

図 2 に示すとおり、CLIV はユニットテスト以前にシステムテストを実施し、プロセスをコンカレント化するモデルである。この改良の効果は 3 つあり、「早期システムテストとサプライヤ間の知識共有」、「プロジェクト内の情報共有と手戻りリスク軽減」、「システムテスト後の修正時間の確保」である。

#### 早期システムテストとサプライヤ間の知識共有

システムテストでは、車両レベルの仕様において、一次サプライヤの理解不足、仕様のグレーゾーンなど、一次サプライヤ間の仕様理解の不一致が顕在化し改善点が明確になる。早期のシステムテスト実施は、自動車メーカー、一次サプライヤ間の情報共有を促進し、改善検討の機会を多く確保することが可能に

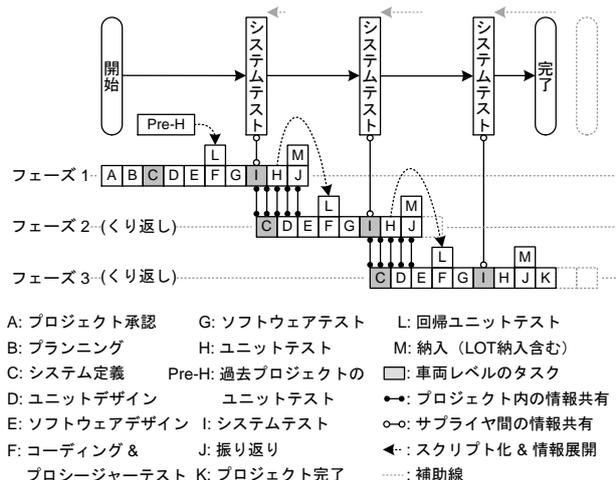


図 3 CLIV のアクティビティ順序と依存関係

なる。

#### プロジェクト内情報共有と手戻りリスク軽減

図 3 に示したとおり、システムテスト以降、現サイクルと次期サイクルの工程が同時進行し、これら工程間にて、プロジェクト内の情報共有が促進される。CLIV では、この機会を活用し手戻りリスク軽減が可能になる。

#### システムテスト後の修正時間の確保

図 3 に示したとおり、システムテストの実施タイミングが早まり、システムテスト後の修正時間の確保が容易になる。特に、最後のシステムテストは、プロジェクト完了前の最後テストであり、システムテスト後の改善時間を確保することが強く求められる。

### 4 CLIV の適用例とその効果

一般に、ユニットテスト以前にシステムテストを実施する場合、システムテストの不具合発生率が高まり、システムテストが十分に実施できないリスクがある。初期の CLIV では、この問題を回避するために、ソフトウェアクリーンルーム理論の「従来の開発パラダイムの移行コンセプト」<sup>9)</sup>を採用し、CLIV の適用対象組織は、プロジェクト上流での仕様管理が十分に行われ、仕様実現段階での品質管理が適切であり、ユニットテストでの不具合発生率が低く抑えられる組織とした<sup>2)</sup>。すなわち、CLIV 適用後、システムテストの不具合発生率の高まりを抑制可能な組織が対象である。

本稿が提唱するモデルでは、さらに、システムテストでの不具合発生率の高まりを抑制する改良を加え、初期のモデルに比べ、モデ

ルの適用可能組織を広くした。このモデルの改良ポイントは 2 つあり、1 つ目は、ユニットテストに対してシステムテストを前に入れることでコンカレント化する改良である。初期のモデルでは、システムテストをソフトウェアテストの前に入れることでコンカレント化していた<sup>2)</sup>。2 つ目は、回帰ユニットテストを導入し、サイドエフェクトバグが原因のデグレードを防止する改良である。これら改良は、初期のモデルに比べ、システムテストの前倒しリスクを軽減する効果をもたらす。なお、初期のモデルでは、回帰ユニットテストは用いられない<sup>2)</sup>。

#### サイドエフェクトバグの防止

CLIV では、回帰ユニットテストを導入し、サイドエフェクトバグの流出を防いでいる。

A 社の 2003 年 1 月から 2005 年 12 月までの 3 年間の TIV モデル 16 プロジェクトにおいて、ユニットテストで発見された欠陥を表 1 に示す。A 社は、日本の自動車メーカー三社の一次サプライヤとして業務活動を行う自動車部品メーカーである。A 社では、サイドエフェクトバグは 31.62%、新機能に関するものは 68.38%、その他は 11.76% である。また、新機能に関するものは正常系と異常系に分類でき、正常系は、16.18%、異常系は、52.20% である。なお、ここでの正常系とは、新機能が正常に動作しているときの振る舞いであり、異常系とは、仕様外の入力に対する振る舞いである。なお、正常系は、システムテストに悪影響を及ぼすリスクがあるが、異常系は、システムテストでは検証しない項目であり、システムテストに影響を及ぼさない。

TIV から CLIV に移行する場合、図 1、図 2 に示すように、システムテストとユニットテストの順番が入れ替わり、システムテストで、サイドエフェクトバグと新機能の不具合が混在し、機能検証に支障をきたすリスクがある。そこで、CLIV では、図 2 に示すように、回帰テストを仕様実現と同時に実施し、サイドエ

表 1 ユニットテストで発見された欠陥 (TIV)

分類	不具合 [%]		ST に影響	
SEB		31.62	あり	
新機能	68.38	正常系	16.18	あり
		異常系	52.20	なし
その他		11.76	あり・なし	

SEB: サイドエフェクトバグ, ST: システムテスト

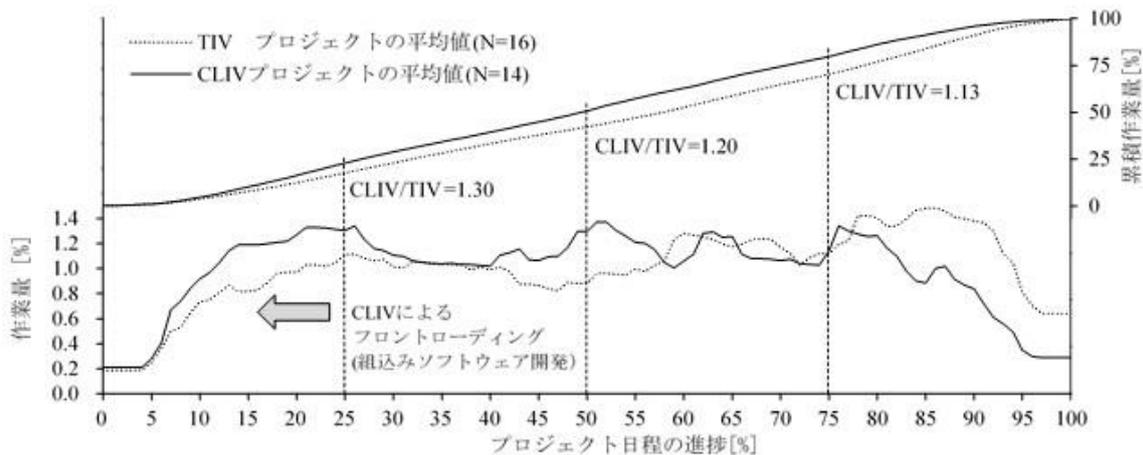


図 4 ソフトウェア開発の作業量の推移 - 「システム定義」から「振り返り」までの作業量  
(作業量[%]:プロジェクト日程の進捗 1%当りの作業量/総作業量)

フェクトバグを未然に防止する施策が施されている。

### フロントローディング化の実現

CLIV 導入後、車載ソフト開発のフロントローディング化が進んだ。累積作業時間の推移を図 4 の上部に示す。このグラフは、観測値が左上に位置するほど、作業がプロジェクト前段にシフトしていることを示している。すなわち、CLIV は TIV に比べて、作業の進行が前倒しになりフロントローディング化が実現されているといえる。さらに、作業進捗度を定量的に比較するために、TIV の進捗を 1 としたときの CLIV の進捗をプロジェクト日程の 25%、50%、75% で比較し、それぞれ、1.30、1.20、1.13 であった。すなわち、これら分析値は定量的に CLIV の作業進捗が前倒しされたことを示している。

次に、プロジェクト日程の進捗 1%当りの作業量の推移を、TIV と CLIV で比較し、図 4 の下部に示した。このグラフは、TIV が、プロジェクト後段の進捗約 78%以降で増加する傾向があることを示し、CLIV が、プロジェクト前段の進捗約 7%から約 32%で増加する傾向があり、さらに、CLIV は、プロジェクト中盤の進捗約 40%から約 57%でも増加する傾向があることを示している。

## 5 まとめ

本稿は、日本の自動車開発における、近年の品質低下問題について、その一因が、ハードウェア設計のフロントローディング化にもなう、車載組込みソフトウェア(車載ソフト)開発の相対的なバックローディング傾向にあるとの視点に立つ。本稿では、車載ソフト

の品質問題の解決策として、その開発プロセスへコンカレント型の反復 V モデル Chain-Linked Iterative V-Model (CLIV)を提唱する。また、CLIV の効果について適用例を通じて検証し、サイドエフェクトバグ防止効果、車載組込みソフトウェア開発プロジェクトにおける作業量のフロントローディング効果を示した。

### 【参考文献】

- 1) 佐伯靖雄：「製品開発組織と開発プロセス：車載組込みシステム開発の設計と調整」, 立命館経営学, Vol. 46, No. 5, pp. 193-219, 2008
- 2) 水上祐治：「車載用組込みソフトウェア開発における米自動車部品サプライヤの日本型開発プロセスへの取組み」, IPSJ-ESS2008 論文集, pp. 199-206, 2008
- 3) Cusumano, M.: "The Business of Software", Free Press, 2004
- 4) 野口祐：「ソフトウェアの経営学」, 森山書店, 1990
- 5) 山田茂, 福島利彦：「品質指向ソフトウェアマネジメント」, 森北出版, 2007
- 6) 藤本隆宏, 武石彰, 青島矢一：「ビジネスアーキテクチャ」, 有斐閣, 2001
- 7) D.E. カーター：「コンカレント・エンジニアリング」, 日本能率協会, 1992
- 8) Kline, S. and Rosenberg, N.: "An Overview of Innovation, The Positive Sum Strategy", National Academy of Sciences, pp. 275-306, 1986
- 9) 二木厚吉, 佐藤武久, 大槻繁, 金藤栄孝：「ソフトウェアクリーンルーム手法 - 高品質ソフトウェア開発パラダイム」, 日科技連, 1997