

6K-6

通言ソフトウェア開発における 共通ソフトウェア開発の問題点とその対策

西本 教人 長谷川 充司 河瀬 浩司 田高 隆志 高倉 聡

富士通九州通言システム (株)

富士通 (株)

1. はじめに

通言ソフトウェア開発において複数の顧客要求仕様に対して、できるだけ共通ソフトウェア（以下は、共通ソフトと呼ぶ）を提供できる様な生産性向上活動を実施している。しかし、度々のエンハンス開発により共通ソフトの構造が複雑化し、共通ソフトの品質低下により顧客先で障害が発生する場合がある。以下に、共通ソフトの品質低下を抑える為に、共通ソフトのインプリメントにおける品質向上施策について検討したのでその内容を紹介する。

2. 背景

現状、通言ソフトウェアの構成は、図1の様、共通部と個別部に分かれている。共通部は、複数の顧客要求仕様に応える様な開発された共通ソフト群で、又、個別部は、特定の顧客要求仕様のみ開発された個別ソフト群である。共通部については、できるだけ複数の顧客に共通ソフトを提供できる様な、ユニットの中で提供機能の要否判定を行うプログラム構成をとっている。しかし、共通部のソフトは度重なるエンハンスによりプログラム構成が変化し、且つ複雑化し、共通ソフト化絡みか要因となる顧客先障害が全体の約20%を占める共通ソフト群（表参照）も存在する。本稿では、複数の顧客要求仕様に対して応え且つ、共通ソフトの品質低下を生じない様にする為の対策について検討した内容を記述する。

3. 共通ソフト開発の品質向上施策

共通ソフトの品質を低下させない施策として、大きく設計/製造工程と試験工程において、下記の2つの施策を開発サイクルに組み込む様考えている。

(1) カプセル判定及びサブユニット化

共通ソフトにおいて機能盛り込みを行う時に、その機能単位に機能番号を持たせ、その機能番号により機能提供の有無判定をカプセル化し、且つ、そのカプセル内に機能盛り込みをサブユニット化（図2参照）し、呼び元ユニットへの盛り込み影響を最小限にする様にする。サブユニット化は、機能提供不要の顧客に対しては、そのサブユニットを未提供で済むことによりメモリ量の削減にも寄与することができる。又、サブユニット化は、その機能に更に機能追加が発生した場合は、更にサブユニット化を進める。

(2) 機能盛り込みユニットの別ライブラリー管理

前述のカプセル判定及びサブユニット化により、既存

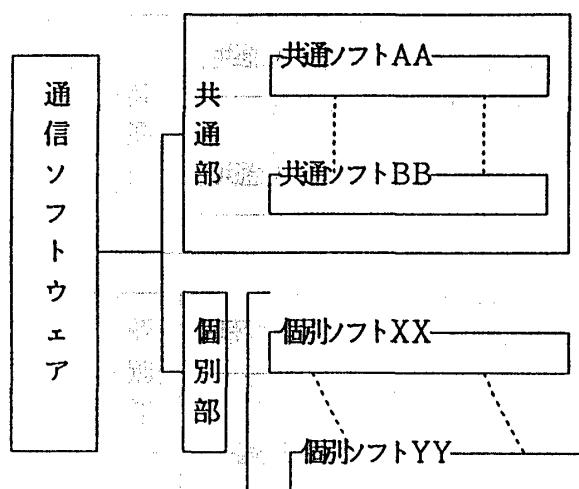


図1 通言ソフトウェアの構成

Problem and Countermeasure for Development of Common Software for Communications Software,

Norihito Nishimoto, Mitsuji Hasegawa, Koji Kawase, Takashi Hidaka

FUJITSU KYUSHU COMMUNICATION SYSTEMS LIMITED

Satoshi Takakura

FUJITSU LIMITED

表1 共通ソフト群の障害盛り込み要因分類

障害盛り込み要因	割合 (%)
機能盛り込みによるシグ	71
共通ソフト化絡みのシグ	17
障害対応によるシグ	8
その他	4
合計	100

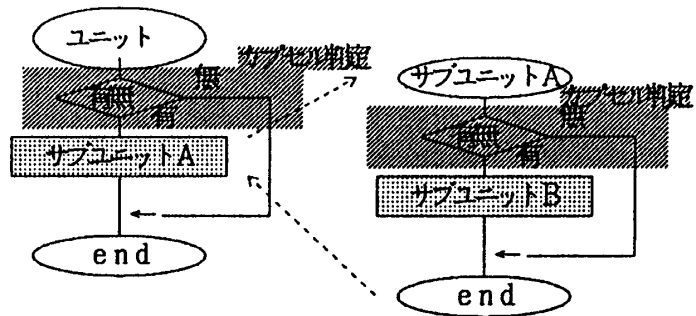


図2 カプセル判定とサブユニット化

のユニットへの影響を抑えると共に、機能盛り込みユニットは、そのユニットの安定性の確認が完了するまでの期間、別ライブラリ化（試験版）し、その期間内は提供を要する顧客に対しては、既存ライブラリのユニット（標準版）を提供する。そのユニットについての盛り込み機能確認試験は、通常の試験工程で実施するが、その後、安定化試験の位置づけで、カプセル化により切分けた機能番号単位の既存機能影響確認試験（表2参照）を実施する。この試験の確認完了後、そのユニットを標準版に移行する様にライブラリ管理する。尚、図3に機能盛り込みユニットの開発サイクルとライブラリ管理の関係を示す。

表2 既存機能影響確認試験一覧

		既存機能				
		A-1	A-2	A-3	A-4	A-5
機能対 象 盛 込 み	A-1	×	有	有	有	有
	A-2	無	×	有	無	無
	A-3	無	無	×	無	有
	A-4	無	無	無	×	無
	A-5	無	無	無	無	×

注 有: 既存機能影響有(試験要), 無 影響無

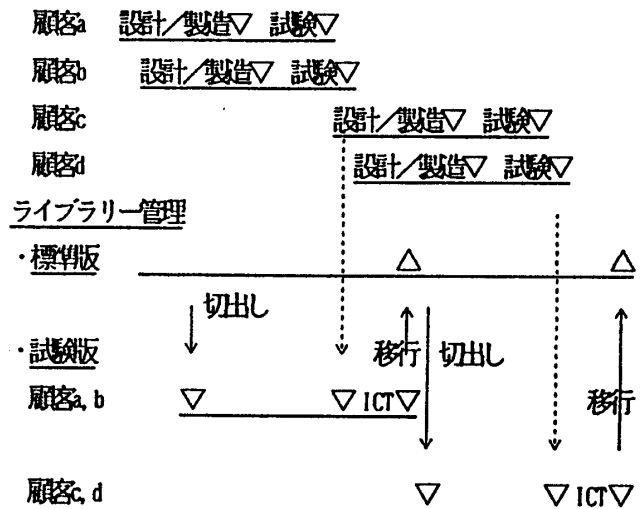
4. 品質向上施策の期待効果

- ① 機能提供不要の顧客に対しても、機能盛り込み開発中及びそれ以降も今まで以上の安定したシステムの提供が可能となる。
- ② 機能盛り込み時の対象ユニットが、サブユニット（部品）りされることにより、今後のインプリメントが容易となり、共通ソフトの品質UPとなる。

5. おわりに

上述の品質向上施策は、現在既に、適用中であるが、今後は、その評価及び共通ソフトのインプリメント工数の最小化を検討していく必要がある。

共通ソフト群の開発サイクル



注 既存機能影響確認試験(Influence Confirmation Test:ICT)

図3 開発サイクルとライブラリ管理の関係