

# TDI組込みソリューションサービス展開とCASEツール ( ~ CASEツール適用によるプロセス改善の薦め ~ )

情報技術開発株式会社 EULシステム営業部

山本 公一朗

## 1. はじめに

このほど弊社はCATS株式会社と事業提携を行い、CASEツール販売において代理店契約を締結いたしました。これを機に弊社では、長年の組込みシステム分野の開発経験を生かした「TDI組込みソリューションサービス」を組込み開発事業分野の大きな柱として、お客様へ展開して参ります。

近年、組込みシステム分野に限らず、様々なシステム開発の世界では開発に従事する技術者の不足や、開発コストの上昇、ソフトウェアの品質問題など、経営者の方を苦しめる問題が多く発生しております。

その中で特に組込み開発の業界はよく3K職場であると表現されることがあり、イメージの悪さから、ますます技術者不足に拍車がかかっている状況です。

このような状況に対し、この分野の研究者や業界の方々によりますと、多くの組込み開発の現場では、昔ながらの開発手法からなかなか脱却できない状況や特定の優秀な人材に偏った要員構成に頼らざるを得ない状況にあり、対策としては開発プロセスそのものを見直して、改善していくことが、今後の組込み分野の発展のためには必要であると分析されています。

そして、改善の手段としてはCASEツールの活用によるプロセス改善が最も効果的であるという、論文や記事を多く見かけます。

弊社がこの様な状況下にある組込み業界のお客様に対し、お客様のプロセス改善のために、どのようなソリューションサービスをご提供できるのかをご説明いたします。

## 2. TDI組込みソリューションの柱

弊社はこれまで、携帯電話・デジタル家電分野における、エンベデッドシステムに組込むソフトウェアの請負開発などを数多く行ってまいりました。

その中で、CMMレベル3の取得、ISO9001の認証取得などを通じて、ソフトウェア品質の向上と開発効率の向上に努めてまいりましたが、その活動にはCASEツールの適用が効果的であり、特に組込み分野における設計工程のCASEツールとして、CATS社の「ZIPC」が有効であることを、モデル開発で検証いたしました。(詳細につきましては本誌の事例紹介にある「ZIPC開発手法と従来開発との比較」をご参照ください。)

CATS社は、日本初のCASEツールベンダーであり、ZIPCを初めとするCASEツールの開発・販売を行ってこられました。

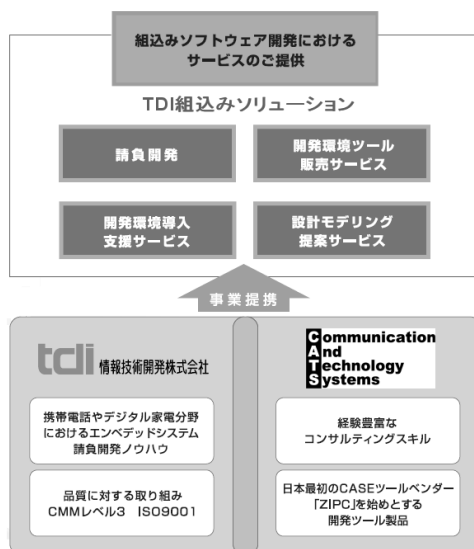


図 1

またCASEツール適用における経験豊富なコンサルティングスキルも持っておられます。

弊社は、そのCATS社と事業提携をさせて頂くことにより、CASEツールの販売代理権とコンサルティング分野などでのご支援を得て

- ・CASEツールによる高品質なソフト請負開発
- ・CASEツール導入支援
- ・CASEツールによる設計モデリングの提供
- ・CASEツールの販売

という、新しいトータルサービスを柱にした「TDI組込みソリューション」を提供させて頂くことで、お客様の様々な課題解決を行って参ります。(図1)

### 3. お客様の課題への対応

既存のお客様へ営業訪問したときや、展示会などにおいて頂いたお客様からは

- ・ソフトウェア品質の向上を効率よく実現したい
- ・人手に依存した開発スタイルを見直したい
- ・CASEツールの導入を検討したいが、適用方法が分からない
- ・既存のソフトウェア資産を有効利用したい
- ・CASEツールは導入したいが初期の導入費用を抑えたい

という声をお聞きます。

TDIはこのようなお客様が抱える課題に対してソリューションを行い、様々なサービスをご提供いたします。

この中で特にCASEツールに関してですが、2005年版組込みソフトウェア産業実態調査報告書の結果(Q2 ソフトウェアツールの課題)\*1の中で、ツール導入の課題になっているのは、「価格が高い」「使いこなすのが難しい」と言う2点がほぼ半数を占めています。

「価格」の点については、TDIがご提案するZIPCファミリーであれば、正しいプロセスでの運用を行うことにより、十分費用対効果が得られると断言できます。

「使いこなすのが難しい」と言う課題については、TDIの組込みソリューションをご活用頂くことで解決できます。

ソフトウェア開発のアウトソーシングには、CASEツールを適用した請負開発サービスのご利用を、CASEツール導入には、CASEツール導

入支援サービス、CASEツールの販売サービスとお客様のニーズにあった、TDI組込みソリューションをご提案いたします。

### 4. CASEツールを適用したソフト開発

TDIへソフトウェア開発をご依頼頂いた場合、請負開発では、ZIPCを活用して、これまで以上に、高品質なソフトウェア開発を行うと共に、ZIPCで作成したドキュメント一式と高品質なソフトウェアをご提供いたします。

また、ご要望に応じて、お客様と一緒に、CASEツールを使ったソフトウェア開発を行ってまいります。さらに、CASEツール導入を効果的にすすめるため、開発フェーズの様々なシーンで、効果的な開発プロセスも提案いたします。(図2)

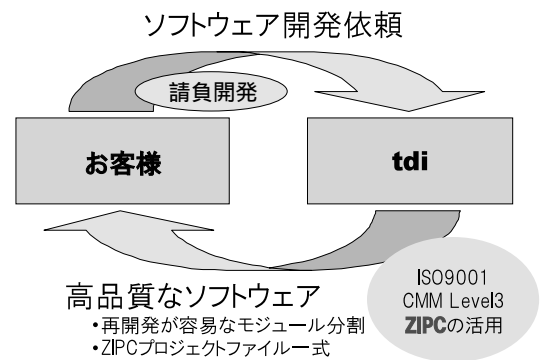


図2

### 5. CASEツール導入支援

まず、CASEツールの導入には、導入計画をしっかりと行うことが必要です。TDIは大きく4つの項目について導入計画を行います。

「ツール適用範囲の検討」

ZIPCを効果的に導入するための適用範囲を検討します。

「CASEツール運用教育」

ツール導入に伴って発生する、運用の仕方・効率の良い開発方法、状態遷移表の作成方法についてアドバイスします。

「開発プロセスの改善」

ツールの効率的運用のため、開発プロセスの改善をご提案します。

「設計サポート」

状態遷移表作成の各フェーズで、必要なサポ

ートをいき、適時、開発現場の皆様のご質問にお答えいたします。

それでは、各々の導入計画について、もう少し詳しく説明して参ります。

(1) ツール適用範囲の検討

ツールの適用範囲は、開発システムの規模・流用資産・開発コスト・現在の抱えている問題点など、各々の開発条件によって異なります。

これらの複雑な条件をもとに、開発への適用部を検討し、適用するフェーズと機能を決定します。

例を示しますと、まず、システムの再開発案件があったとします。機能2と機能3が追加部で、機能1と機能4は、従来の機能をそのまま引き継ぎます。このような場合、システム開発の全ての局面にZIPCを適用することも考えられますが、流用部の品質が高く、適切なモジュール化ができていれば、再開発をさせ、追加機能の2と3にだけ、ZIPCを適用して開発することもできます。(図3)

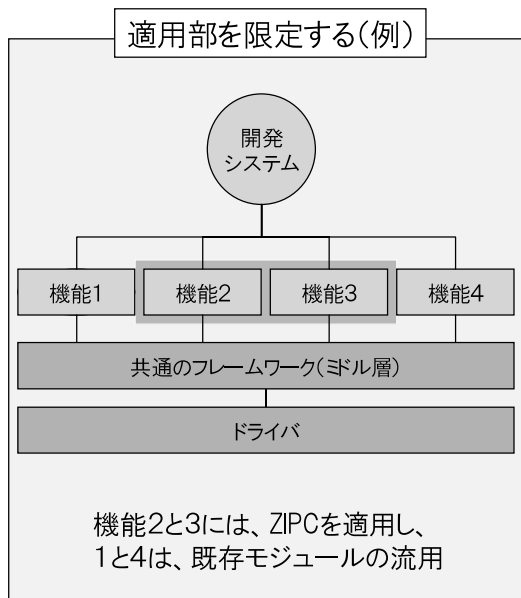


図3

また、ZIPCのどんな機能を使用するか？ということも、検討が必要です。

例えば、状態遷移表を作成するのに、Excelを使用していたならば、ZIPCを状態遷移表エディ

タとして使用することで、作業効率を向上させることができます。

状態遷移表エディタとしてだけでなく、シミュレータ機能も使用することで、ドキュメントの動的検証を行い、設計品質を大幅に向上させることができるようになります。そして、Cコードジェネレート機能を使えば、状態遷移の振る舞い部のソースを自動生成することができ、コーディング時間とテスト時間が減り、ドキュメントとCソースコードの整合性を保った開発を行うことができます。(図4)

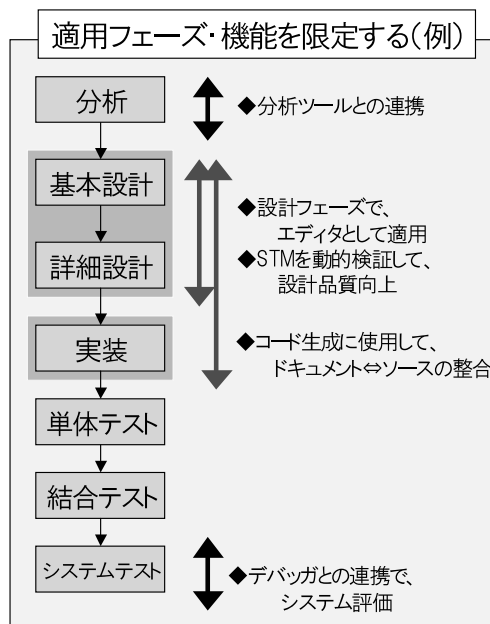


図4

このように、適用部・適用機能を明確にすることで、ツール適用を計画的に実施することができます。「どのような機能部(全体なのか?それとも特定の機能なのか?)に適用するか?」、「どの開発フェーズでの利用が効果的か?」をその都度、検討する必要があります。

(2) CASEツールの運用教育

CASEツールは、導入すればすぐに効果を発揮するようなツールではありません。CASEツールを効率的に運用するためには、開発メンバーが、ZIPCなどのツールに対する正しい知識を習得し、効率的運用ができるようになる必要があります。例えば、ZIPC導入が決まったとき、

皆さんは、ここにあるような不安や問題を抱えるのではないのでしょうか？（図5）

- ・ ZIPCというツールが使いこなせるか？  
（ 難しくないか？ ）
- ・ 状態遷移表ってどうやって書くのか？
- ・ 階層化するには、どうしたらいいのか？
- ・ 効率のよい開発の進め方は？



図 5

これらの内容は、OJTやOffJTなど、ケースバイケースの指導を受けることで、効率よく習得することができます。

正しい知識を習得し、少しでも早くCASEツールを活用した開発を始められるよう、お手伝いいたします。

### ( 3 ) 開発プロセスの改善

従来の「ウォーターフォール開発」と「ZIPCを使用したウォーターフォール開発」を比較した場合、「従来のウォーターフォール開発」では、次工程に進む前に設計書のレビューで設計品質を確保してきました。

しかし、これでは設計工程で作りこんでしまうバグを取り除くことは難しく、下流の評価工程である結合テストやシステムテストになってから不具合を検出し、大きな手戻りを発生させてしまいます。

一方、ZIPCを使用した場合は、各設計書をシミュレーションすることで動的に検証することができ、設計品質が格段に向上します。

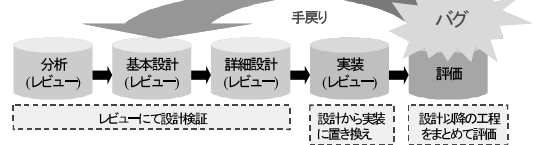
上流工程の検証で品質の高い設計書を作成し、さらにその設計書から作成したソースコードには、設計書の修正を伴うような大きな問題は含まれません。結果として、結合テストやシステムテストをスムーズに終わることができ、手戻りの発生も低く抑えることができます。（図6）

また、ZIPCのCソースコード生成機能を使用することで、設計書から実装までをシームレスに開発することができ、工程間のコミュニケーションロスによる不具合も無くなります。

CASEツールを使うことにより、このようにして、開発プロセスが改善されます。

TDIは実際の開発現場の内側から支援を行うことで、お客様のプロセス改善をより確実に、より強固に行わせて頂くことが可能です。

#### ■従来のウォーターフォール開発



#### ■ZIPCを使用したウォーターフォール開発

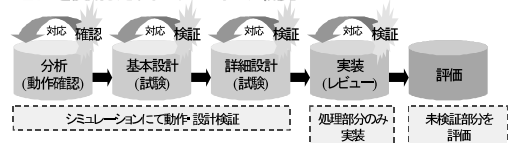


図 6

### ( 4 ) 設計サポート

設計サポートでは状態遷移表を作成するという作業を3つのフェーズに分けて考えます。（図7）

#### ・ 状態遷移表作成準備

このフェーズでは、これから作成する状態遷移表は、どのようなものなのか、設計に盛り込む要素（イベントと状態）とその目的を明確にします。

実はここが最も難しく、この部分が明確になっていないと、正しい状態遷移表を作成することができません。

#### ・ 状態遷移表作成・検証

状態遷移表を記述する・検証するというのは、ZIPCを使った作業です。

この作業の中で、どのように表現すれば良いか？うまく書けるのかといったことから、ZIPCの使い方まで、いろいろな疑問・質問が出てきます。

それらの質問に迅速にお答えし、開発が滞ることが無いよう、しっかりとサポートを行います。

#### ・ 状態遷移表レビュー

レビューを2つのポイントで行います。

1つめのポイントは「振る舞いが正しく表現できているか?」です。

これは、仕様を熟知する必要がありますので、基本的に、関係者によるレビューが必要です。

2つめのポイントは「状態遷移表が正しく記述できているか?」です。

状態遷移表の作成に矛盾がないか? 無理な記述方法がとられていないか? 次のステップに進むことができる状態になっているか? など、そのフェーズでの粒度が状態遷移表として妥当性があるかをレビューします。これには、状態遷移表について、正しい知識が必要です。

TDIは、状態遷移表を作成する際、これらの問題を解決するためのサポートを行います。

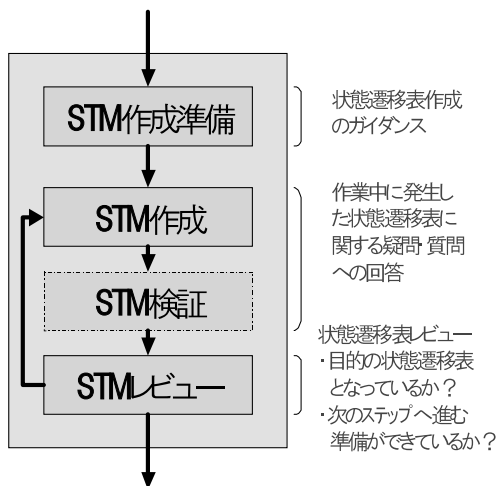


図 7

## 6. ZIPCを用いた開発プロセス

ここではTDIがお客様に提案する開発プロセスとはどのようなものなのかを説明させていただきます。

CASEツールであるZIPCを導入することにより提供されるのは「開発手法」と「開発プロセス」です。「開発手法」としては「拡張階層化状態遷移表設計手法」をサポートしています。

この開発手法については、書籍も販売されており、CATS社のHPからも説明書をダウンロードすることが可能ですので、そちらをご覧ください。開発手法では定義されていない、ZIPCを有効的に使うことによってもたらされる「開発プロセス」がありますので、その開発プロセスについて概要を説明いたします。

その開発プロセスとは「モデルベース開発(モデル駆動型)」と「スパイラルでの成長型モデル開発」であり、この2つがZIPCを使った開発のキーワードになります。

ZIPCでの開発プロセスを簡単に追うと、次のような順序になります。(図8)

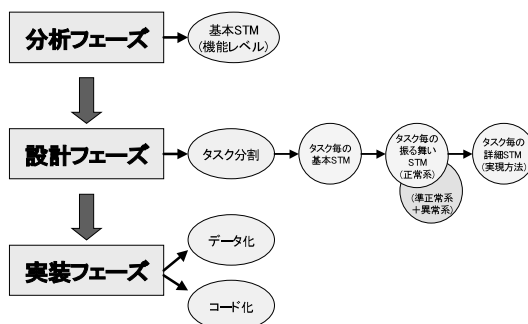


図 8

まず、基本状態遷移表を作成したあと、その状態遷移表を元に、タスク分割を行い、タスク毎に、基本状態遷移表から、詳細化を進めます。

実装フェーズでCソースコードジェネレートを行う場合は、データ化とコード化を行います。

ドキュメントとして作成した状態遷移表の処理部を関数化したり、データ設計を行ったり、実装したりといった、従来開発と同じような詳細設計~実装を行い、日本語から英文字への置換情報を作成して、コード出力を行います。

分析・実装フェーズでは、仕様書から、基本状態遷移表を作成し、その状態遷移表をもとに、タスク設計を行って、タスク毎の機能状態遷移表を作成します。

それらを次第に詳細化することで、詳細な状態遷移表を作成します。(図9)

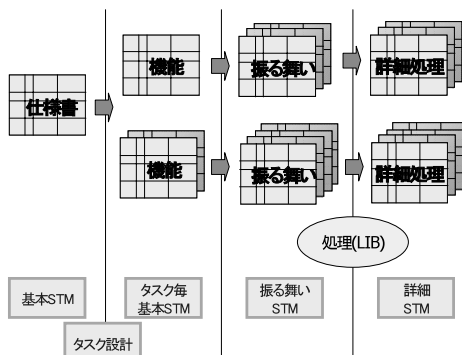


図 9

実装フェーズでは、設計書としての状態遷移表から、コード生成を意識した状態遷移表へ修正します。具体的にはデータ化で処理を関数化し、モジュール設計とコーディングを意識した内容に置換します。さらにコード化で日本語を英文字に置換します。(図10)

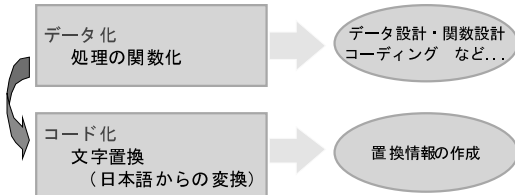


図10

成長型モデル開発のメリットは、抽象化レベルから、詳細レベルへ段階的に落とし込むことで、機能ごとに整理された設計となり、結果としてモジュール化が促進されます。

また、このようにして作成したシステムは、機能追加を行うフェーズが明確になり、類似や派生開発などでの再利用性が向上します。

同じ開発手法を行うのに、Excelを使用して行うこともできますが、修正変更が難しく、シミュレーションもできない状態遷移表では、作業効率が著しく低下します。

よって、CASEツールを導入することが良い方法となるのです。

## 7. ZIPC導入でのFAQ

この章ではZIPC導入の局面で、一般的に不安に思われるのではないかと課題をFAQの形で紹介します。

まず、良くあると思われるQuestionとして以下の4つを想定してみました。

- ・ZIPC適用が有効な処理部とは？
- ・ZIPC適用範囲のバリエーションは？
- ・ツールの使用方法が難しいか？
- ・ZIPCの使用実績は？

以下に各々について説明します。

ZIPC適用が有効な処理部とは？

ZIPCを適用する場合に、大変有効な処理とそうでない処理があります。

ZIPCは、状態遷移表を用いたCASEツールのため、基本的に状態遷移が発生しない機能部(ドライバなどの下層部や、シーケンス処理を行う機能部など)にCASEツールを適用しても、効果が出にくいこととなります。

状態遷移がある機能部とは、一般的に、ドライバ層より上位の層です。(図11)

アプリケーション層やミドルウェアのような上位層の機能モジュールは、現在の状態によって振る舞い(処理)が変化するので、その振る舞いを状態遷移表で設計することは大変効果的です。

また、ドライバ層のように、状態を持たないような処理にもZIPCは適用できます。表にする意味があまりなく、不向きではありますが、表記の統一やソースコードの自動生成などプロセスの一貫性を保つ意味で適用するという点では効果的です。

ZIPC適用範囲のバリエーションは？

ZIPCの適用範囲とその方法について、8つの例を挙げてみました。

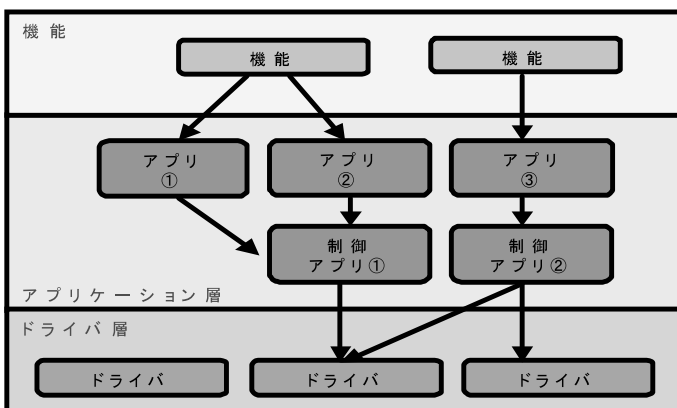


図11

### ZIPC適用 有効箇所

状態(条件)によって、振る舞い(処理)異なるため、状態遷移表が有効である。

### ZIPC適用 可能箇所

基本的にどの層においても、適用することが可能である。

(1) 既存システム整理のため、全てに適用する  
 既存システムの完全な置き換えにZIPCを適用する方法です。コストと時間がかかりますが、仕様変更の繰り返しにより、その品質を保つことができなくなってしまったようなソフトウェアに対して、極めて有効です。

(2) 既存システムの一部を整理するために部分適用する  
 仕様変更に影響を及ぼす機能部を事前に整理する目的で適用する場合に有効です。

(3) 既存システムの追加新機能についてのみ適用する  
 追加部の機能を周辺の影響を受けずに開発する方法で、最も現実的な適用例です。

(4) 新規開発システムに対して全適用する  
 新規案件について、特に制約が無ければ、最も効果的な開発方法です。

(5) 新規開発システムに対して部分適用する  
 新規案件に対して、初期導入費用を抑え、ZIPCの適用効果を確認するのに良い方法です。

(6) ツール開発に対して適用する  
 テストツールの開発などにZIPCを適用することで、ツール群の開発を容易に行うことができます。

(7) 機能分析にてVIP機能のみを適用する  
 VIPを使ったシミュレータを作成し、機能分析・仕様FIXを行うために、適用することもできます。

(8) 設計工程にてシミュレーション機能のみを適用する  
 ZIPCを状態遷移表の品質向上・作業効率向上のために使用するが、従来の開発スタイルを変更したくないなどの場合に有効です。

以上の例は、ZIPC適用における、ほんの一例です。どのような適用方法が最も効果的であるかは、既存システムのおかれている状況、実際の開発内容や、スケジュール、予算などによっ

て左右されます。

TDIでは長年の組込み開発経験とCASEツール適用事例経験、CATS社の支援などにより、お客様にとって最適なCASEツール適用範囲の提案と導入サポートをソリューションサービスとして提供させて頂くことができます。

ツールの使用方法が難しいか？

ZIPCで状態遷移表を作成することは、難しいことではありません。

ツールの使い方よりも、「状態遷移表を上手く書く」ということが難しいのではないのでしょうか？

状態遷移表を作成する場合に、巨大で複雑な状態遷移表になってしまい、検証するのも大変な状態で、收拾がつかなくなってしまった状態を皆様もご経験されたことがあると思います。

ZIPCがサポートする状態遷移表は、階層化することにより、可読性の高い状態遷移表を維持しようとするものです。

基本状態遷移表で、最も基本的な要件の振る舞いを記述し、基本状態遷移表をもとに段階を経て、状態遷移表を詳細化していきます。(図12)

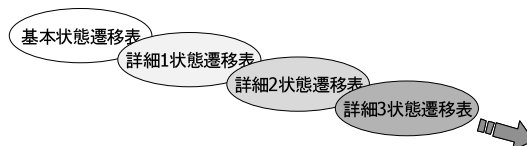


図12

このとき、モジュール化・階層化といった、手法を取り入れ、状態遷移表の巨大化を防ぎながら可読性の高い状態遷移表を作成して行きます。

ZIPCの使用実績は？

これについては、この場であまりご説明する内容はありません。なぜなら、本誌をご覧いただければ、一目瞭然だからです。

バックナンバーもありますので、是非この機会に本誌だけでなく、バックナンバーもお読み頂くことをお勧めします。

また、本誌には、弊社のモデル開発事例を掲載して頂いており、その中でZIPC適用による効果を説明しております。(本誌ユーザ事例紹介の「ZIPC開発手法と従来開発との比較」を参照)

尚、下記の図はそのとき収集したメトリックスの一部を表とグラフにまとめたものです。(図13)

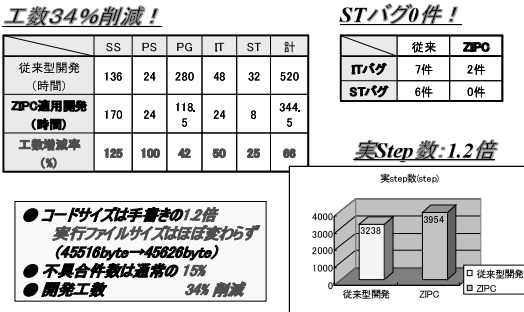


図13

## 8. おわりに

今までの説明をおさらいする意味とCASEツールであるZIPCの導入効果を改めて認識して頂くために、ZIPC開発における主な効果とよくある不安に対する答えを表にしました。(表1)

### ■ZIPC開発 効果

No.	効果	内容
1	品質向上	<ul style="list-style-type: none"> <li>状態遷移表設計により、設計の漏れ・抜けが減少する</li> <li>上流工程にて設計の動的検証が可能になるため、上流工程にて品質確保ができる</li> </ul>
2	自動コード生成	<ul style="list-style-type: none"> <li>設計書からの自動コード生成が可能であるため、PG生産性が向上する</li> <li>設計書とソースコードは、常に整合性を保つことができる</li> </ul>
3	自動テスト検証	<ul style="list-style-type: none"> <li>設計書(シーケンス)から、自動テスト検証ができる</li> <li>シミュレーションログからシーケンスを生成し、再帰テストの実行ができる</li> </ul>

### ■ZIPC開発 不安

No.	不安	内容
1	初期導入コスト	<ul style="list-style-type: none"> <li>費用対効果を検討する必要がある</li> <li>⇒各々のケースにて検討が必要だが、十分に効果がある認識です</li> </ul>
2	設計モデリング教育	<ul style="list-style-type: none"> <li>ZIPCはあくまでツール(道具)であるため、状態遷移表によるモデリング手法が重要</li> <li>⇒TDIにてしっかりサポートいたします</li> </ul>

表1

CASEツールを導入すれば、確実にプロセス改善の効果が得られます。

しかし、導入に至るまでも、導入後にも様々な不安や問題があるのも事実であると思います。

お客様がお持ちになる、そのような多くの不安や問題に対して、TDI情報技術開発は確実にお応えし、お客様のプロセス改善に対して、ZIPCファミリーの導入とそのサポートによるソリューションサービスをご提供させて頂くことができます。

CASEツールの導入をお考えのお客様、プロ

セス改善をお考えのお客様や現状のプロセスに不安や問題を抱えているお客様は、是非ともCATS社のZIPCファミリーをお選び頂くとともに、CASEツールによるプロセス改善を強力にサポートできるTDI情報技術開発にソリューションをお任せ頂きたいと思ひます。

CATS社と情報技術開発の今後の活躍にご期待ください。

\*1) 平成17年1月 経済産業省商務情報政策局

情報政策ユニット 情報処理振興課の調査結果より

### お問い合わせ窓口

情報技術開発株式会社  
EULシステム営業部  
山本 公一朗  
TEL : 06-6201-7781  
FAX : 06-6201-7741  
E-Mail : kumikomi@tdi.co.jp