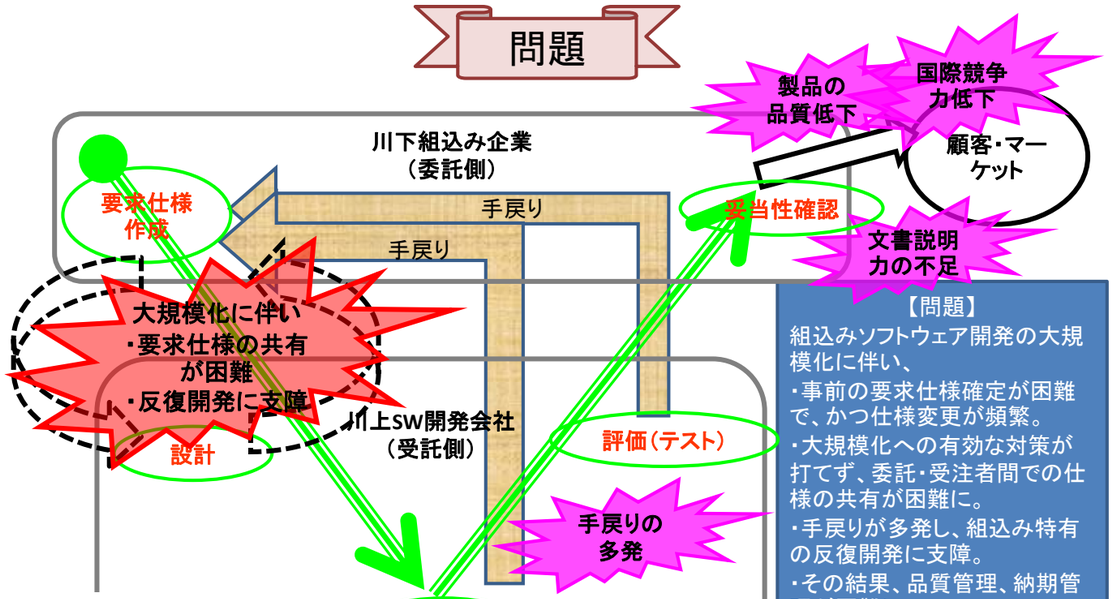
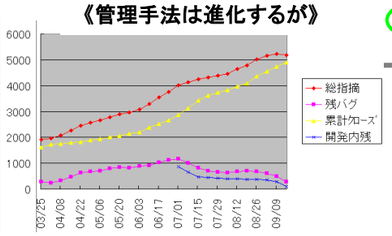
問題

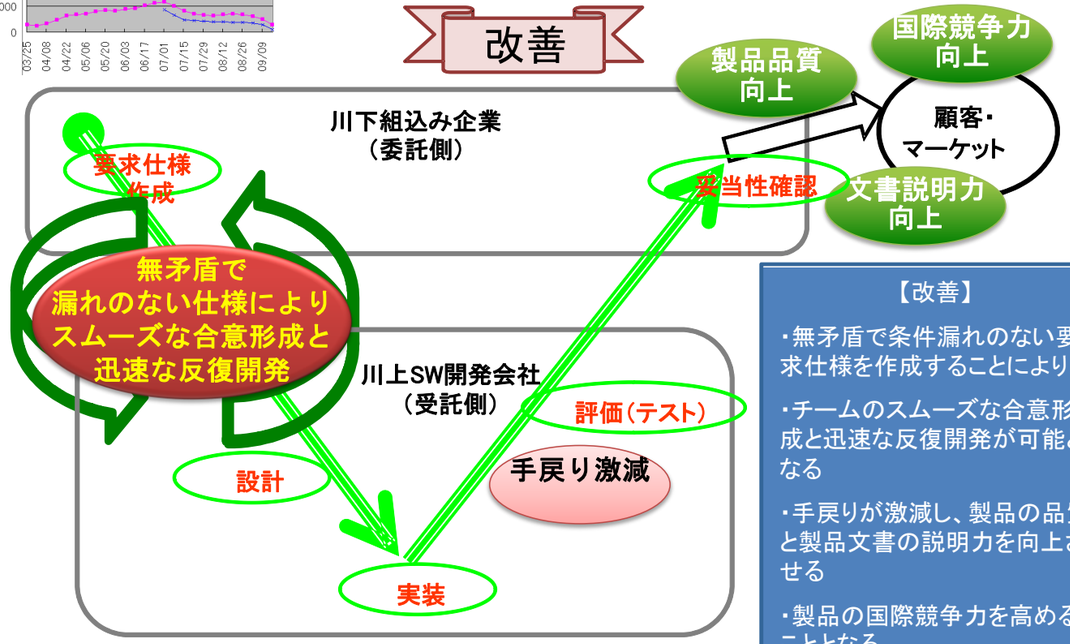


【問題】
組込みソフトウェア開発の大規模化に伴い、
・事前の要求仕様確定が困難で、かつ仕様変更が頻繁。
・大規模化への有効な対策が打てず、委託・受注者間での仕様の共有が困難に。
・手戻りが多発し、組込み特有の反復開発に支障。
・その結果、品質管理、納期管理が困難に。
・市場で不具合問題が発生しても、十分な説明ができない。

《管理手法は進化するが》



改善



【改善】
・無矛盾で条件漏れのない要求仕様を作成することにより、
・チームのスムーズな合意形成と迅速な反復開発が可能となる
・手戻りが激減し、製品の品質と製品文書の説明力を向上させる
・製品の国際競争力を高めることとなる

株式会社ジェーエフピー
〒020-0063 岩手県盛岡市材木町 2-26
http://www.jfp.co.jp/

JFP metaphor Lab.

本ソフトウェアの開発は「ものづくり中小企業試作開発等支援補助金」に採択されています。

要求仕様記述ツール

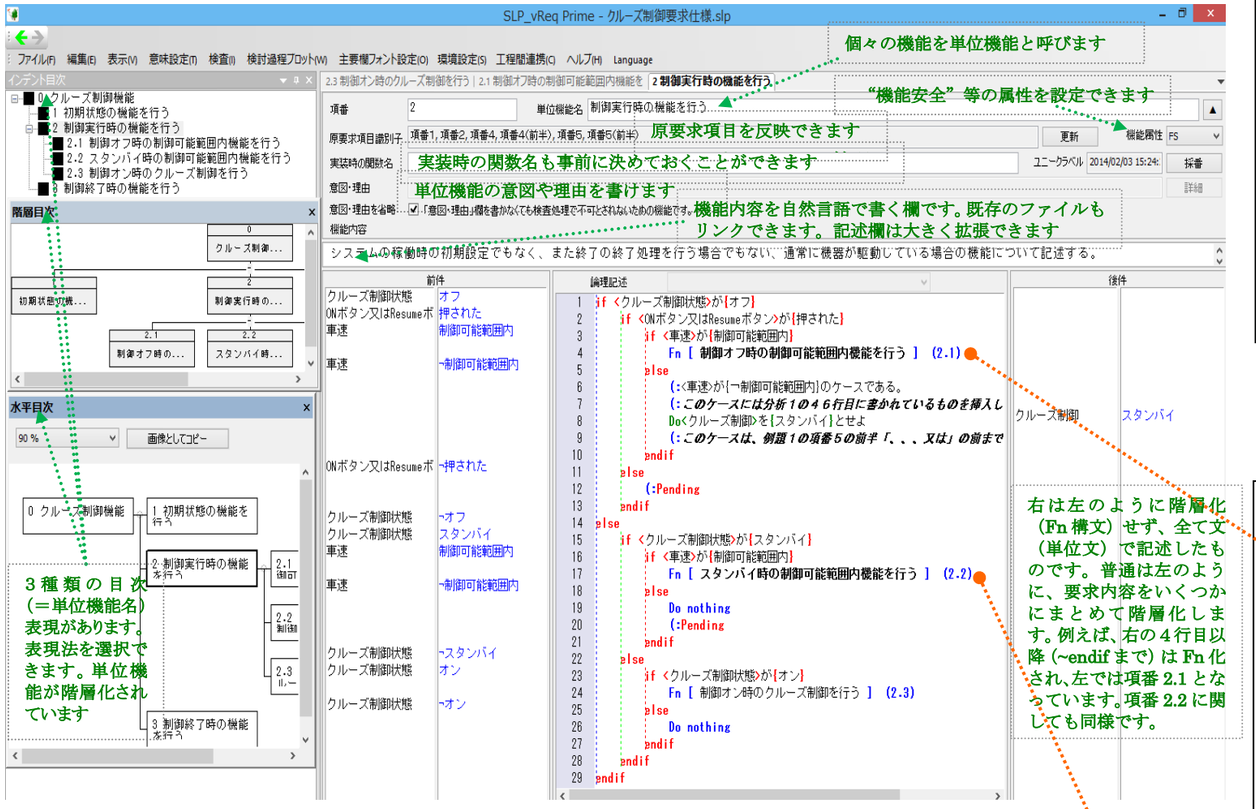
SLP

Logical

本ソフトウェアは要求仕様を
簡単な**文法**で正確に
記述するツールです

要求仕様書が**論理整合性と一意性**を持ちます
要求から実装まで要求の**トレーサビリティ**を実現しています
操作と運用に**高いユーザビリティ**を発揮します

JFP metaphor Lab.



《SLP の基本的な考え方》

ソフトウェア要求仕様書はプログラムが実現すべき内容からなる文の集まりです。文は内容の説明文も含みますが、最終的には実現すべきことが「～ならば、～せよ、そうでないならば、～せよ」という条件付き命令文と、「～せよ」だけの無条件命令文からなります。「～」は文で、主語や目的語（対象物）と述語（対象物の状態）からなります（単位文）。「ならば」と「そうでないならば」を条件接続詞と呼ぶこととします。すると、要求仕様書は単位文と条件接続詞で記述される、といえます。

《SLP 構文》

1．**2つの基本構文**と**3つの派生構文**と、**2つの参考文**からなります。これらを **SLP 構文**と呼びます。基本構文は、例示の **Do 文**と **if 文**です。派生構文は基本構文を基に、単位機能を表す **Fn 文**、if 文の派生としての **switch 文**、繰り返しの **loop 文（for 文、while 文）**です。参考文はコメント文と改行文です。

2．文は対象物(主語や目的語)とその対象物の属性（述語）とで書かれます。これを**単位文**と呼びます。対象物を「メンバー」と呼び、属性を「状態」と呼びます。状態名は陳述的述語や命令的述語で書かれます。たとえば「実行ボタンが ON であるならば」は「if <実行ボタン>が{ON}である」と記述します。“if” に対しては必ず “else” が付属され、if-else-endif の構造を持ち、if-else-endif は複数の入れ子構造となります。メンバーの状態を設定（命令）する時には、「Do<実行ボタン>を｛OFF｝せよ」と記述します。

3．要求仕様内容は**階層的**に表現できます。大きな文書も内容の親子・兄弟関係が明確にされます。階層化される単位を**単位機能**と呼び、SLP 構文で論理的に記述されます。これにより文が矛盾無く記述されているか、また用語が文書全体の中で統一されているかを検査できます。このことは、製品企画、要求分析者、開発者、テスト者等の思い違いを減少させ、開発予算の最適化の前提を形成します。

4．SLP は対象物に着目して記述しますので、モデル構造を備えています。メンバー名がモデルのオブジェクトとなります。状態名がオブジェクトの属性であり、if-else-endif の記述が機能仕様に相当します。

仕様書例

（“SCADE” のテキストより）

【クルーズ制御仕様】

- **クルーズオン制御**

1. クルーズ制御状態がオフで，ON ボタン又は Resume ボタンが押され，車速が制御可能範囲内で，アクセルペダルとブレーキペダルが踏まれていない時
2. クルーズ制御状態がスタンバイで，車速が制御可能範囲内で，アクセルペダルが踏まれていない時

- **クルーズオフ制御**

3. 初期状態は必ずオフである
4. ブレーキペダルが踏まれるか，OFF ボタンが押された時

- **クルーズスタンバイ制御**

5. クルーズ制御状態がオフで，ON ボタン又は Resume ボタンが押され，車速が制御可能範囲内でない又は、アクセルペダルが踏まれて，ブレーキペダルが踏まれていない時
6. クルーズ制御状態がオンで，車速が制御可能範囲内でない又は，アクセルペダルが踏まれている時

実際にお試しください。ダウンロード→www.jfp.co.jp/slp/

上の例は仕様としてはよく書かれているものと思われます。しかし SLP で実際に要求分析をしてみると、クルーズ制御に関する常識的なことを知らないと、よく書けないことに気づきます。このことは、仕様分析者は、仕様の意味をすべて知らなければならないことを意味しているように思われます。しかし、SLP は仕様の「意味」を知らない人でも、「形式」（SLP の構文）に従い記述することで、「意味」（ドメイン知識）をしっかりと無駄なく（正確に、効率的）把握して行くけしように支援すます。SLP は初心者の仕様分析ツールでもあります。

```
if <クルーズ制御状態>が{オフ}
  if <ONボタン又はResumeボタン>が{押された}
    if <車速>が{制御可能範囲内}
      if <アクセルペダルとブレーキペダル>が{¬踏まれている}
        Do<クルーズ制御>を{オン}せよ
        (:上のifの4つは例題の項番１の構文化。
      else
        if <アクセルペダル>が{踏まれている}
          if <ブレーキペダル>が{¬踏まれている}
            Do<クルーズ制御>を{スタンバイ}とせよ
            (:このケースは項番５の後半を指す。
          else
            Do nothing
            (: <ブレーキペダル>が{踏まれる}ケー
            (: ス(項番4の前半)。なお「踏まれる」
            (: と「踏まれている」の差異は、必要
            (: であれば、SLPでは、forやwhile、
            (: loop構文などで表現できる。
          endif
        else
          Do nothing
          (: <アクセルペダル>が{¬踏まれる}かつ
          (:<ブレーキペダル>が{踏まれている}のケース。
        endif
      endif
    else
      Do<クルーズ制御>を{スタンバイ}とせよ
      (:<車速>が{¬制御可能範囲内}のケース。
      (:このケースは項番５の前半（,,, 車速が制御
      (:可能範囲でない）を指す。
    endif
  else
    (:Pending
  endif
else
  if <クルーズ制御状態>が{スタンバイ}
    if <車速>が{制御可能範囲内}
      if <アクセルペダル>が{踏まれていない}
        Do<クルーズ制御>を{オン}せよ
        (:上のelseとifの3つは項番２を指す。
      else
        Do<クルーズ制御>を{オフ}とせよ
        (:項番4の<アクセルペダル>が{踏まれている}を指す。
        (:Pending
      endif
    else
      Do nothing
      (:Pending
    endif
  else
    if <クルーズ制御状態>が{オン}
      if <ブレーキペダル>が{踏まれる}
        Do<クルーズ制御>を{オフ}とせよ
        (:このケースは項番４の前半。
      else
        if <ONボタン>が{押された}
```

“¬” は否定記号

文はメンバー名<>と状態名｛｝から成ります。メンバー名や状態名は、以前に記述したものの一覧表から選択できます。助詞、助動詞を付記できます。

上記例の**自然言語**（通常の文）で書かれた仕様書を左のように記述することによって、**論理整合性**（論理無矛盾、条件網羅性）と**一意性**（用語統一、意味解釈が１つ）を獲得できます。

記述は SLP 構文で**単位文**（主語/目的語＋述語）を書くことで行います。また複数の単位文をまとめ、１つの**単位機能**とすることができます。単位機能は**階層的**に表現（記述）されます。

記述は構文のメニュー選択で行います。記述可能な構文をガイドします。さらに以前に書いたメンバー名や状態名を再利用できます。用語の揺らぎ防ぎます。

SLP 構文は if-else-endif 構文により**条件網羅性**を保証します。また、単位文の主語や目的語が持っている述語を検査することにより、同じ主語や目的語が同じ述語の否定形を持っていないかチェックします（**形式矛盾**の指摘）。さらに、**単位文の事前登録機能（辞書）**により、主語や目的語が持ってはならない述語のチェックをします（**意味矛盾**の指摘）。

SLP では単位文の指す意味を**単位意味**と呼びます。意味の数から仕様の規模を測定できます。また SLP での**レビュー**は単位意味ごとに確認できますので、レビューにありがちな細かな部分の**確認ミス**を防止できます。

開発には**仕様変更**はつきものといわれますが、SLP では仕様変更は「**単位文の用語を変えること**」や「**単位文どうしの接続関係を変えること**」と形式（構文）的な観点から論じ、変更を形式的な数で把握できます。SLP ではどの単位文も仕様書全体に渡って関連づけられますから、用語や接続関係の変更による**影響範囲**も把握が容易で、また仕様変更の**差分**管理も行うことができます。

SLP は if-else-endif 構造を**決定表**に変換し、条件の一覧性を示すことができます。また**テストの仕様原案**を自動作成できます。

要求項目は固有の識別子を付与されますので、要求仕様書、設計書、プログラムソースまで**トレーサビリティ**が実現されます。

SLP 構文で記述した文書を**通常文**に逆変換し、自然言語に近い表現をすることができます。