

顧客からの要求と開発の進め方

土屋 泰, 坂本 直史

ここでは、本特集を通して開発する制御システムの顧客からの要求事項を説明する。開発対象とするのは、CANプロトコルを利用した自動車ドアの表示システムである。また、どのような手順に沿って開発するのかを説明する。
(編集部)

1. 今回のプロジェクトの課題

本特集では、車載機器や産業用制御機器、医療機器など、多方面で使用されている通信プロトコルであるCAN (Controller Area Network) を使って、自動車のドアを制御する分散制御システムを開発します。

なお、この事例はあくまでも架空のものであり、実際の車載アプリケーション・システム開発とは異なります。あらかじめご了承ください。

● 顧客の要求は突然に

営業Aさん：こんにちは。

顧客Kさん：やあ、いらっしゃい。ちょうど今、開発してもらいたいシステムがあるんだ。自動車のドアの制御なんだがね(図1)。

Aさん：ドアの制御？ 開け閉めするんですか？

Kさん：いや、開け閉めは実際には人間がやるんだが、そ

れぞれのドアを個別のボード(マイコン基板)で制御するシステムを開発してほしいんだ。なに、いきなり実車に実装してくれ、なんて言うわけじゃない。まずはボード上で状態を表示できるようにして、デモンストレーションとしてボードを動かしてほしいんだ。

Aさん：ボード上で表示、ですか...？

Kさん：なに、デモンストレーションだから、7セグメントLEDとか、そんなもんでいいよ。

Aさん：(??)

● 頭の中のアイデアを説明する

Kさん：いいかい、すべてのドアはそれぞれ個別に開閉できるし、ロックも制御できる。まあ普通はそうだろう？ で、エンジンをかけたら、あらかじめ設定しておいた速度で自動車が走るんだ。「走る」と言っても、デモンストレーション・システムでは、取り込んだ値を表示してくれればいいよ。

Aさん：はい。ドアの開閉やロック/アンロックは遠隔操作するんですか？

Kさん：いや、そうじゃなくて... 説明がまずかったかな。

ドアは手で開け閉めするんだ。制御ボードは基本的にドアの状態を認識しているだけだ。ただ、デモンストレーションとしては、ボードに対してドアの開閉を入力できるようにしてほしい。

ロックも基本は手動だな。でも、運転席からはすべてのドアをロックできる必要があるな。それから、自動車がある程度以上の速さで走っていると、オートロックがかかるようにしてほしい。

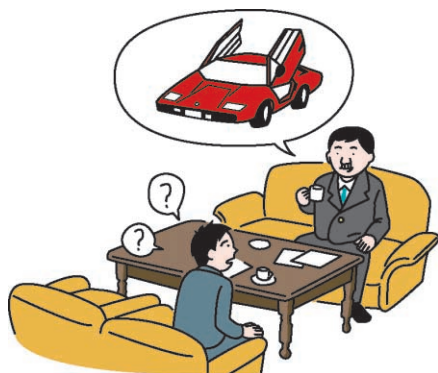


図1 顧客の要求を聞く

Aさん：はあ。(とりあえずメモを取る)

...エンジンはどこでかけるんですか？

Kさん：決まってるだろ，運転席だよ．ああ，ここでは運転席の制御ボードだな．

ほかのボードは，それぞれ自分のドアの開閉状態とロック状態を表示してくれればいい．けれど，運転席のボードは，エンジンが動いているか停止しているかといった状態を表示してほしい．それから，走っているときは車速もね．

Aさん：はい...

Kさん：あ，それからドアの数なんだが，4ドアと2ドアの車の両方に対応できるように作っておいてほしいんだ．いちいちプログラムを変えなきゃならんようなのは面倒だからね．

Aさん：分かりました．で，ドア間はどのような通信にしましょうか？

Kさん：デモンストレーションだから，CANバスで通信してくれれば，後は任せるよ．

Aさん：そうですか...．では後日，技術の者を連れてきますので，その者に詳しいお話をいただければと...

● 文書にまとめる

そのようなわけで，技術者であるGさんがKさんのもとへ赴き，聞き出した内容を「自動車ドア関連制御システム(デモンストレーション・システム)の開発」(図2，p.50)としてまとめました．これが，今回の課題の大本となる顧客の要求になります．

2. 開発モデル(開発プロセスとは)

ソフトウェア開発モデルとは，読んで字のごとく「ソフトウェアを開発する際にどのように開発していくかという手本，テンプレート」です．「ソフトウェア開発モデル」(以下では簡単に「開発プロセス」と呼ぶ)と書くとは難しいもののように感じますが，ソフトウェアを開発するときに沿うべき手順と考えてもらえれば分かりやすいと思います．

ここでぜひとも知っておいてもらいたいのは，開発プロセスは「与えられるもの，決まったもの」ではなく，自分たちの開発の効率を上げていくための「たたき台」とであるということです．

開発プロセスは，過去に成功した開発(ベスト・プラク

ティス)に学んで無理，無駄，ムラをできるだけ減らし，より良い開発を行うために使用するものです．ただし，組み込みシステムの場合は対象となるシステムの特性が幅広いため，代表的な開発モデルと会社で決められた開発モデルがぴったり合うということはほとんどありません．実際に適用していくと，「うまく合わないところがある」などの問題点や，「こうしたらもっと良くなりそうだ」といった改善点が出てくると思います．そのような問題が明らかになったら対策を考え，その対策が有効であるかを実開発で確認し，開発の特性に合うように開発プロセスを徐々に改善していけばよいのです．

代表的な開発モデルとして，ウォーター・フォール型や反復(スパイラル)型などの手法があります．これらの手法のどれを基本として採用するかは皆さんのプロジェクトで決めることですが，ここは，組み込みソフトウェア開発でよく使用されているウォーター・フォール型モデルを使って，開発をどう進めていくかの流れを説明していきます．

● 開発モデルと開発工程

図4にウォーター・フォール型の開発モデルのイメージを示します．ウォーター・フォール(water fall)は「滝，落水」を意味します．滝は一方向にしか水が流れません．そのため水(開発)が上位工程の器を満たせば(上位工程での開発が終われば)水が落ちて次の工程の器を満たし始める，つまり上位工程から順番に開発していこうという考え方です．

この図では，開発工程を「要求分析」，システムのアーキテクチャを決定する「基本設計」，プログラムの構造を決め

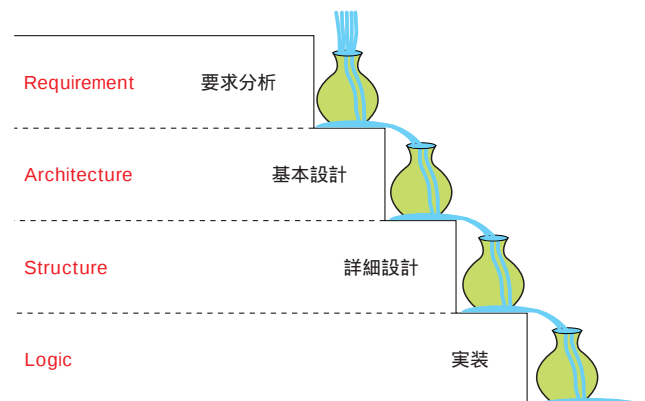


図4 ウォーター・フォール型の開発モデル

上位工程から順番に開発していこうという考え方である．なお，工程の呼び方は組織や分野によって異なることがある．