Prologue

杉浦 英樹

組み込みソフト開発の世界へようこそ

ここでは，本特集全体の導入部として，厳しい制約のもとでより良いシステムを構築する組み込みソフトウェア開発が日本の文化とよくマッチしていることを説明する．また，設計技術のプロフェッショナルとアマチュアの違いや製品の品質，およびその品質を実現するための考え方について紹介する．（編集部）

皆さん，組み込みソフトウェア開発の世界へようこそ．

本稿は，新たに組み込みソフトウェア開発を担当される方に，組み込みソフトウェアの世界を理解してもらうための特集です．

第1部（第1章～第3章）では，組み込みソフトウェアを開発するために「必要なコト」，「必要なモノ」を明らかにした上で，組み込みソフトウェアの開発プロセス（開発手順）に従って，新人エンジニアの皆さんに組み込みソフトウェア開発のコツを理解してもらうためのサンプル・プログラムの作成を試みます．また，開発プロセスを通して行われる実務の課題，さらには組み込みソフトウェアの開発環境の課題を紹介します．

第2部（第4章～第6章）では，まず，開発プロセスを実践する際の考え方を紹介します．工数見積もりやスケジュール管理，構成管理など，製品開発を進める上で必要な開発管理について解説します．さらに，開発手法や管理手法など，最新の技術情報を提供し，皆さんが開発を進めていく中で，どのような技術に着目すればよいのかを示します．

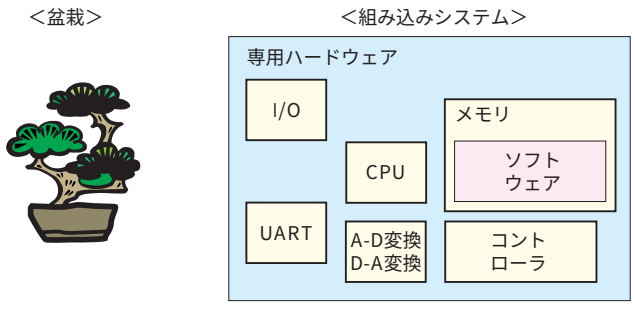

図1　盆栽と組み込みシステムの共通性

ひと鉢の中に自然のシステムを組み込み，自然を楽しむ盆栽は日本独特の文化．組み込みシステムは，盆栽と同じように，専用基板の上に必要な部品を配置し，顧客の期待を実現する．日本が組み込みシステム開発を得意とするのは，このようなユニークな文化（制約を楽しみ，制約の中に何かを見いだす思想）**とも関係するのではなかろうか．**

● 箱庭や盆栽の文化と似ている組み込みソフト開発

一口に組み込みソフトウェアの開発といっても，実際に開発するソフトウェアは，制御対象ごとに異なります．このことが組み込みソフトウェアの特徴とも言えます．そして，組み込みソフトウェアの開発ほど，日本の文化にマッチするものはないのではないかと筆者は考えています．

日本は，四方を海に囲まれ，大陸と行き来しにくい環境で文化をはぐくんできました．このような状況があったからこそ，箱庭や盆栽といった，制約を楽しむ文化が生まれたのではないかと思います．

箱庭や盆栽は，一つの箱や鉢の中に，石，土，木を配置します．限られた養分の中で植物をはぐくむと，居ながらにして見事な自然のメカニズムや季節を愛でることができます．これは，「効率を上げるためなら浪費もいとわない」といった合理性に基づく発想では創出できない，美しいバランス感覚の上に成り立った一つのシステムだと思います．

目的のためには手段を選ばず，大きなコンピュータ・パワーや大容量のメモリ資源を消費するエンタープライズ系ソフトウェアやデータ系ソフトウェアの場合と異なり，組み込みソフトウェアの開発は，とても多くの制約を前提としています．このことは，組み込みソフトウェア開発がまさに日本の文化にマッチしていることの証しだと思います（**図1**）．

● プロフェッショナルであることの意味

組み込みシステムは，とても多くの制約の中で要求を実

KEYWORD ── 組み込みソフトウェア，箱庭，盆栽，QCD，品質，洞察力，コミュニケーション能力，PDCAサイクル，ものづくり，職業技術者

現します．こうしたシステムは，企業の基幹情報サービスを構成するような大規模ソフトウェアの開発と比べると，規模が小さく，一見アマチュアでも作れそうな気がします．しかし実際の組み込みシステムの開発現場は，数多くの誇り高いプロフェッショナルが支えています．

組み込みソフトウェア開発に限ったことではありませんが，プロ・アマの違いは“QCD”にあると言えます．つまり，要求されるQ（Quality ；品質），C（Cost ；コスト），D（Delivery ；納期）を達成するのがプロです（**図2**）．

一方，趣味の電子工作などでは，こうした要件は軽んじられる傾向にあります．特に製品ソフトウェアの品質については，例えば，制御対象となる組み込みシステムが，工場で製造される部品の精度ばらつきの影響を受ける場合，その影響を考慮した開発が求められます．

● 組み込みソフトウェアの品質とは

組み込みソフトウェアの品質について，もう少し考えてみましょう．そもそも，品質とは何でしょう．品質にはさまざまな定義がありますが，「組み込みソフトウェアの品質」には，「要求を満足すること」という簡単な言葉が当てはまります．ここで重要なのは「要求」です．

組み込みシステムの利用者が，組み込みシステムに期待することすべてを「要求」ととらえましょう．要求には，明文化されるものと，そうでないものがあります．機能や性能，信頼性，安全性など，さまざまな要求が明文化されます．しかし，文書になっていることだけ実現できれば品質を達成できるかというと，そうではないのです．要求者が文書にすることができなくても，「システムはどう振る舞うのか」，「通常，考慮する範囲を超えたとき，システムはどう振る舞うべきなのか」ということを要求者とともに考え，合意し，明文化する必要があります．

組み込みソフトウェア開発者には，このような洞察力とコミュニケーション能力が求められます．まずは，自分のチームに関係するさまざまな人たちとよく話し，彼らがどのような仕事をしているのかを理解するところから始めてください．その上で，自分たちの仕事の「品質」は何かということを検討し，定義しましょう．

● 計画・実施・チェック・改善のサイクルを回す

もう一つの重要なポイントは，“PDCA”の実践です．本特集の第2部（第4章～第6章）でも述べますが，P（Plan ；計画）を立てて，D（Do ；実施）し，C（Check ；チェック）し，A（Act ；改善）し，次のPを立てて…と連鎖するのがPDCAサイクルです．このサイクルを正しく回すことができれば，どのような仕事もうまく進めることができます．もちろん，開発を通してさまざまな問題が起こりますが，起こった問題でさえ，経験としてプラスになっていくのがPDCAサイクルの良いところです．

図2　プロとアマの違い

プロとアマの違いはQCDのレベルの違いにある．ゴルファにたとえるなら，スコアの違いや飛距離の違いがQ（品質）に，道具やトレーナにかける投資の違いがC（コスト）に相当する．出場したら必ずラウンドするのがプロで，アマは不調を押してまでラウンドしない．この違いがD（納期）．それ以外にも，コース戦略の違いやスコアを維持するための練習計画の違いなどがある．フォーム改良の活動では，PDCAサイクルを素早く回すのがプロ．プロのPDCAサイクルに比べてアマのPDCAサイクルは回転速度が遅いと言える．

筆者の知る限り，PDCAサイクルがうまく回っている開発チームは必ず成功しています．逆もまた然りで，特に問題なのは，回しているつもりになっているチームです．PDCAを，次のPDCAにつなげられないこともしばしばです．つまり，本来はPDCAPDCAP…と回しますが，PDCA / PDCA / P…となり，改善が次の計画に反映されないのです．これではせっかく良いことを経験しても，結果が生かされません．そして，うまくいったことは再現できず，失敗したことは再発してしまいます．このようなことをなくすには，CAP（チェック・改善・計画）の部分に着目して活動するのが効果的です．

PDCAサイクルは，ものづくりにおけるすべての考え方の基本になります．QCDを効果的に達成することを目標とする職業技術者には「必須の哲学」として，PDCAサイクルを実践することを心がけてください．

すぎうら・ひでき　富士ゼロックス（株）

News Flash ―― Cypress，赤外線利用の動き検出装置向けPSoC開発キットを発売
Cypress Semiconductor社は，焦電型赤外線（PIR）を利用した動き検出装置の開発キット「CY3286A-PIRMOTION」を発売した．PSoC（Programmable System-on-Chip）マイコンを搭載する．照明の制御や自動ドア開閉装置，無線カメラなどに使えるという．