

組込みソフトウェアに関する知識体系

組込みソフトウェア管理者・技術者育成研究会

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

著作権について

- この知識体系は、組込みソフトウェア管理者・技術者育成研究会（SESSAME）が著作権を所有しています。営利目的とした複製、利用をされる場合にはあらかじめSESSAMEの事務局から承諾を受ける必要があります。
- 知識体系を利用したことによって生ずるいかなる損害に関しても、組込みソフトウェア技術者・管理者育成研究会（SESSAME）は一切責任を負いません。
- 知識体系に関するご意見・ご提言・ご感想・ご質問等がありましたら、組込みソフトウェア技術者・管理者育成研究会（SESSAME）事務局までE-Mailにてご連絡ください。

- 組み込みソフトウェア技術者・管理者育成研究会事務局：

〒113-8656

東京都文京区本郷7-3-1 東京大学大学院 工学系研究科 化学システム工学専攻

飯塚研究室

E-mail: sessame@blues.tqm.t.u-tokyo.ac.jp

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

組込みソフトウェア管理者・技術者育成研究会

- 東京大学大学院工学系研究科の飯塚教授が発足
略称 SESSAME (セサミ): Society of Embedded Software Skill
Acquisition for Managers and Engineers
- 組込みソフトウェアを世界の競争の中で将来のコアコンピタンスに
→ 組込みソフトウェアの技術者や管理者が数万人レベルで不足
- 早急な育成が必要
→ 育成するためのカリキュラムの整備
→ その基になる方法論・ツールの開発や研究

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

日本でのソフトウェア開発の現状

- エンジニアの勤勉さと生真面目さに立脚
→ 海外の組込み製品と比較して3倍働いて2倍の品質を作り込む
- ソフトウェア工学的な設計技術やプログラミング技術
→ 実用レベルとはほど遠い
- 一貫した戦術を適用する文化がない
→ 常に出たところ勝負
- 品質保証チームと開発チームの溝
- 経営者、上級管理者が組込みソフトの開発管理技術を持たない
→ 現場が根性や努力のみで問題解決

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

日本のソフトウェアに対する危惧と対処戦略

- 日本ソフトウェア産業の現状
 - 創造性型産業において米国に劣る
 - コスト競争型産業でアジア諸国に負ける
- 組み込みソフトウェア
 - PCソフトウェアのような「そこそこ品質」ではダメ
 - より高品質、高信頼性
 - ハードウェアとの協調という点で日本に向き
- 組み込みソフトウェア開発
 - 日本の特性を生かして世界の競争の中で
 - コアコンピタンスとして伸ばすのが提案する戦略

Copyright © 2002組み込みソフトウェア技術者・管理者育成研究会

現状打破のために

- やはり基本は人材の育成
 - リアルタイム組み込みソフトウェアの教育カリキュラム？
 - ・ マイコン、リアルタイムOS、C言語といった個別はあるが、
 - リアルタイム組み込みトータルではない
 - リアルタイム組み込みソフトウェアの設計手法
 - ・ 昔の大型機、今のPC用ソフトウェアの設計手法はあるが、
 - リアルタイム組み込みをターゲットとした試みは少ない



人材育成のための教育カリキュラム
リアルタイム組み込みソフトウェアに適した設計手法の検証

Copyright © 2002組み込みソフトウェア技術者・管理者育成研究会

現状打破のための人材育成

- 育成すべき人材

実際にリアルタイム組込みソフトウェアを開発する担当者

- 一般的なソフトウェア工学の技術や知識は有効
- 一般業務用ソフトウェアではマイナーなドメインへの深い知識

ex. ハードウェアの制御ドメイン知識
 リアルタイムプログラムの開発手法

組込み開発の問題を理解し適切な環境を用意できる管理者

- 人材育成のための教育カリキュラム用知識体系の整備

組込みソフトウェアのための知識体系

- 現場に即した実用的な知識を体系化 知っているべきこと！

- ソフトウェアのライフサイクル

▣ 体系化

▣ システム分析と要求定義

▣ システム設計

▣ ソフトウェア設計

▣ プログラミング

テスト

運用・保守

システム分析と要求定義（1）

< 知識の分類 >

< 項目 >

調査
技法 / 図法 / 記法
ドキュメント
分析

類似システムの調査
適用分野（ドメイン）の調査
適用分野（ドメイン）の理解
対象製品のマーケティング

システム分析と要求定義（2）

< 知識の分類 >

< 項目 >

調査
技法 / 図法 / 記法
ドキュメント
分析

ブレインストーミング
コンセプトモデリング
タイミングチャート
ユースケース
DFDなど
・データフロー・ダイアグラム
・状態遷移図
・状態遷移表

システム分析と要求定義（3）

< 知識の分類 >

調査
技法 / 図法 / 記法
ドキュメント
分析

< 項目 >

ドキュメントの種類
・機能定義
・制約条件
ドキュメントの書き方

フィージビリティスタディ
リスク分析
その他の分析
・ボトルネックの調査
・ベンチマークテスト
・開発対象の識別

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

システム設計（1）

< 知識の分類 >

ハードとソフトの切り分け
冗長設計
技法 / 図法 / 記法
インターフェース定義
OSを使うか使わないか
適合すべき規格
使用環境
アナログとデジタル
ビジネス要件からの設計要件

< 項目 >

切り分けの視点
・スケジュール ・お金
・制約条件 ・保守性
・製造容易性
人間系とハード/ソフトのトレードオフ
・人間の行う役割を明確にする
・フォールトトレラント/フェイル
セーフ/フルプルーフ/リンプホーム
をシステムのどの要素が担うか
・処理するデータの性質

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

システム設計（２）

< 知識の分類 >

ハードとソフトの切り分け

冗長設計

技法 / 図法 / 記法

インターフェース定義

OSを使うか使わないか

適合すべき規格

使用環境

アナログとデジタル

ビジネス要件からの設計要件

< 項目 >

冗長設計を実現する手段

- ・フェイルセーフ
- ・フェイルオペラティブ
- ・単一点故障除去

冗長設計の構成

- ・デバイスレベルの冗長構成
- ・プロセッサ・レベルの冗長構成
 - ・コールドスタンバイ
 - ・ホットスタンバイ

冗長設計の程度

システム設計（３）

< 知識の分類 >

ハードとソフトの切り分け

冗長設計

技法 / 図法 / 記法

インターフェース定義

OSを使うか使わないか

適合すべき規格

使用環境

アナログとデジタル

ビジネス要件からの設計要件

< 項目 >

SADT

ブロックダイアグラム

- ・ソフト側から見れば
データフロー図のマッピング
- システム記述言語

マンマシンインターフェース

外部機器とのインターフェース

内部のインターフェース

システム設計（４）

< 知識の分類 >

ハードとソフトの切り分け
冗長設計
技法 / 図法 / 記法
インターフェース定義
OSを使うか使わないか
適合すべき規格
使用環境
アナログとデジタル
ビジネス要件からの設計要件

< 項目 >

選択肢
選択基準
・ライセンスコスト
・リアルタイム性能, etc
腹のくくり方

安全規格
ノイズ
輸出を考えるか否か

システム設計（５）

< 知識の分類 >

ハードとソフトの切り分け
冗長設計
技法 / 図法 / 記法
インターフェース定義
OSを使うか使わないか
適合すべき規格
使用環境
アナログとデジタル
ビジネス要件からの設計要件

< 項目 >

耐久性 気圧
熱 温度保証範囲
消費電力 電源電圧

ノイズ対策
計測システム

台数 コスト 保守体制
供給安定性（特殊な部品を使うか）

ソフトウェア設計（1）

< 知識の分類 >

設計のための概念

設計で理解しておくべき

ソフトウェアの概念

構造化設計

OOD

設計補助技法 / 図法 / 記法

ハードやOSに関する情報

品質 / 信頼性

設計検証

< 項目 >

抽象化 / 詳細化

アーキテクチャ設計

階層構造（ヒエラルキー）

モジュール化

・ 結合度 / 凝縮度

・ 情報隠蔽

・ ライブラリ / 部品化

再利用

デザインパターン

ファクタリング

設計のルール

言語 / ライブラリ / ソフトウェア部品選択

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

ソフトウェア設計（2）

< 知識の分類 >

設計のための概念

設計で理解しておくべき

ソフトウェアの概念

構造化設計

OOD

設計補助技法 / 図法 / 記法

ハードやOSに関する情報

品質 / 信頼性

設計検証

< 項目 >

イベントドリブン

エクセプション / 例外

ドライバ / ハンドラ

タイマ / ウォッチドッグ

タスク間通信

レスポンスタイム

ボラタイル

並列 / 並行処理 次のページで説明

データ管理

ヒープとガベージ

パーシステンシー（不揮発性）

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

ソフトウェア設計（3）

< 知識の分類 >

設計のための概念

設計で理解しておくべき

ソフトウェア的概念

構造化設計

OOD

設計補助技法 / 図法 / 記法

ハードやOSに関する情報

品質 / 信頼性

設計検証

< 項目 >

並列/並行処理

- ・ シングル/マルチタスク
- ・ シングル/マルチプロセッサ
- ・ タスク/プロセス/スレッド
- ・ プライオリティ
- ・ 資源の競合
 - ・ クリティカルセクション
 - ・ セマフォによる回避
 - ・ セマフォの種類
 - ・ デッドロックの発生
- ・ 割り込みとポーリング

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

ソフトウェア設計（4）

< 知識の分類 >

設計のための概念

設計で理解しておくべき

ソフトウェア的概念

構造化設計

OOD

設計補助技法 / 図法 / 記法

ハードやOSに関する情報

品質 / 信頼性

設計検証

< 項目 >

技法

- ・ データ指向設計技法
 - ・ データフロー設計
 - ・ DeMarco法
 - ・ Yourdon法
 - ・ Hatley/Pirbhai法
 - ・ データフローからプログラム構造への変換
 - ・ その他の構造化設計手法
- 図法 / 記法
- ・ 階層構造図
 - ・ バブルチャート
 - ・ PAD / HCP

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

ソフトウェア設計（5）

< 知識の分類 >

設計のための概念
設計で理解しておくべき
ソフトウェアの概念
構造化設計
OOD
設計補助技法 / 図法 / 記法
ハードやOSに関する情報
品質 / 信頼性
設計検証

< 項目 >

技法
・ Shlaer/Mellor法
・ ROOM法
・ OCTOPAS法
図法 / 記法
・ UML
・ シーケンス図
・ クラス図
・ コラボレート図
・ ステートチャート

ソフトウェア設計（6）

< 知識の分類 >

設計のための概念
設計で理解しておくべき
ソフトウェアの概念
構造化設計
OOD
設計補助技法 / 図法 / 記法
ハードやOSに関する情報
品質 / 信頼性
設計検証

< 項目 >

状態遷移図 / 状態遷移表
RMA (Rate Monotonic Analysis)
プロトタイピング
OSの仕様やAPI
メモリサイズ / ROMサイズ
メモリ種別
メモリマップ
I/Oマップ
ハードウェアの要件
ハードウェア関係の資料一式

ソフトウェア設計（7）

< 知識の分類 >

設計のための概念
設計で理解しておくべき
ソフトウェア的概念
構造化設計
OOD
設計補助技法 / 図法 / 記法
ハードやOSに関する情報
品質／信頼性
設計検証

< 項目 >

フォールトトレラント
自己診断
フェイルセーフ/フルブーフ
信頼性解析
テスト容易性設計
セキュリティ

デザインレビュー
・ピアレビュー
・ウォークスルー
・フォーマルレビュー
バリデーションとベリフィケーション

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

プログラミング（1）

< 知識の分類 >

言語に依存しないもの
言語に依存するもの

< 項目 >

プログラミングで意識すべき
ハードウェアに関する情報
コーディングで理解しておくべき
ソフトウェア的概念
ソフトウェア的に気を付けるべき部分
チューニングと最適化
デバッグ
コードレビュー/コードインスペクション
品質の確保
構成管理

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

プログラミング（2）

< 知識の分類 >

言語に依存しないもの

言語に依存するもの

プログラミングで意識すべき

ハードウェアに関する情報

コーディングで理解しておくべき

ソフトウェア的概念

< 項目 >

CPUに依存する情報

- ・ 開発環境に関する情報
- ・ マイコンのアーキテクチャ

CPUに依存しない情報

- ・ 周辺ハードウェア仕様
- ・ 実行のインフラに関する知識

アルゴリズムとデータ構造

ポーリングと割り込みの実装

リエントラント性

再帰呼び出し 禁止！

プログラミング（3）

< 知識の分類 >

言語に依存しないもの

言語に依存するもの

ソフトウェア的に

気を付けるべき部分

チューニングと最適化

< 項目 >

コンパイラのくせやバグ

移植性

リソース管理

トップダウンとボトムアップの混合

多重割り込み

チューニング

- ・ 速度 / メモリ

最適化

- ・ 再利用性 / 移植性 / 省電力

プログラミング（４）

< 知識の分類 >

言語に依存しないもの

言語に依存するもの

デバッグ

コードレビュー /

コードインスペクション

< 項目 >

スタブやドライバ

デバッグ容易コーディング

デバッグ

実機を意識したテストパターンや

テストデータ（ベクタ）

バグの分析／切り分け

コードレビューの種類

・コードインスペクション

・ウォークスルー

コーディング規約

コード品質最適化

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

プログラミング（５）

< 知識の分類 >

言語に依存しないもの

言語に依存するもの

品質の確保

構成管理

< 項目 >

テストファースト

カバレッジ

テストの実行環境

データフロー分析/テスト

コメントの量と質

動的な品質

バージョン管理

Copyright © 2002組込みソフトウェア技術者・管理者育成研究会

プログラミング（6）

< 知識の分類 >

言語に依存しないもの

言語に依存するもの

構造化系言語に依存する項目

アセンブラに依存する項目

〇〇系に依存する項目

< 項目 >

手続き型プログラミング

- ・ フローチャート
- ・ 構造化プログラミング
- ・ GOTO文（論争）

- ・ モジュール分割
 - ・ 構造化チャート
 - PAD / HCP

- ・ 変数スコープ

C言語に依存する項目

プログラミング（7）

< 知識の分類 >

言語に依存しないもの

言語に依存するもの

構造化系言語に依存する項目

アセンブラに依存する項目

〇〇系に依存する項目

< 項目 >

アセンブラの得意な部分

- ・ クリティカルセクション
- ・ スーパーバイザーコール
- ・ キャリー付き演算
- ・ 積和命令

〇〇系言語の原則

- ・ 多重継承はあまりやるな
- ・ 深い継承はしない
- ・ Cohesion に気を付ける
- ・ 演算子オーバーライドは禁止
- ・ 関数を使うな

ご参加をお待ちしてます

・組み込みソフトウェア管理者・技術者育成研究会（SESSAME）は、組み込みソフトウェア技術者や管理者を育成するためのカリキュラムの整備、そしてその元になる方法論・ツールの開発に関する研究を行っております。

・前述したように、我々の感触では、組み込みソフトウェアの中級レベル以上の管理者・技術者が数万人以上必要であり、早急に養成する必要があるものと確信しています。それゆえに、組み込みソフトウェアに関心を持っている方々に、一緒に議論してほしいと思っております。詳細はSESSAMEのホームページの趣意書や組み込みソフトウェア管理者・技術者育成研究会へのお誘いをご覧ください。

SESSAMEのホームページ：<http://blues.tqm.t.u-tokyo.ac.jp/esw/>

・入会はどなたでもお気軽に下記までご連絡ください。

nishi@tqm.t.u-tokyo.ac.jp