

平成23年度

モデルベース開発人材育成研修
(システム検証研修)

シラバス

(NO,6 講座シラバスにつきましては只今準備中です)



公益財団法人 ひろしま産業振興機構
Hiroshima Industrial Promotion Organization

7	システム工学概論
担当講師	広島市立大学 大学院情報科学研究科 教授 大場 充
概要	<p>自動車系組み込みソフトウェア開発の内外における動向を、特に標準化の視点から理解することを目的とします。自動車製品の品質は、それを構成する個々の部品の品質と、それらを統合するプラットフォームの品質によって決定づけられます。また、自動車製品における品質の概念は、組み込みソフトウェアの比重が重くなるにつれ、従来のISO9000の品質マネジメントから、さらに踏み込んでソフトウェア開発そのものを問題にする Automotive Spice(ISO/IEC15504の自動車への適用)や、組み込みソフトウェアの検証法も要求するような安全性を機能の視点で見直す IEC61508などが標準化されつつあります。また、技術的には車載ネットワーク(FlexRayなど)や基本ソフトウェアにオープンソースソフトウェア(OSS)などを応用した開発が求められつつあります。そのような環境下において、我々はどのようにモデルベース開発を導入・展開すべきかを考えるために必要な基礎知識や課題を幅広く理解します。</p>
使用教材	資料を配布
講義の流れ ポイント	<ol style="list-style-type: none"> 1. ソフトウェアの本質と難しさ <p>なぜ、ソフトウェアの開発は難しく失敗が多いのかを考える。「ソフトウェアは見えない」とよく言われるが、「見えない」とはどのようなことなのか、そして見えないことで、何が問題になるのかを、リスクの視点から分析する。また、携帯電話などの組み込みソフトウェア開発の失敗が、メーカにどのような損害をもたらし、なぜそのような問題が起きたのかを、ソフトウェア工学の視点から分析する。</p> 2. 組み込みソフトウェアとは <p>組み込みソフトウェアと一般のソフトウェアは、どこが、どのように違うのかを理解する。そして、一般のソフトウェアはどのように開発されているかを、大規模ソフトウェアの例で学ぶ。また、一般の大規模ソフトウェア開発では、どのような問題が発生するのかを、みずほ銀行のオンラインシステム事故などの事例から学ぶ。</p> 3. 組み込みソフトウェアはどのように開発するのか <p>一般の大規模ソフトウェアと同じように、組み込みソフトウェアにも、典型的な開発手順がある。例えば、米国政府のNASAが開発しているスペースシャトル搭載コンピュータを制御するソフトウェアは、どのような人々が、どのような手順で開発してゆくかを学び、自動車に搭載されるコンピュータを制御するソフトウェアは、どのように開発されるべきかを学ぶ。</p> 4. ソフトウェア開発プロセスとは <p>ソフトウェアを開発する手順のことを一般にソフトウェア開発プロセスと呼び、典型的なプロセスの組み方をプロセスモデルと呼ぶ。そのようなモデルの中でも最も一般的で、スペースシャトル搭載コンピュータのソフトウェア開発に適用されているウォーターフォールモデルについて学び、その弱点を理解する。さらに、その弱点を解消したいいくつかの新しいモデルについて理解し、個々の現場で、どのように自動車に搭載されるコンピュータのソフトウェアを開発すればよいかを考えるための基礎知識を得る。</p>

5. なぜソフトウェア開発プロセスを問題にするのか

ISO9000 や ISO14000 においては、「プロセスが重要」と言われる。それは、プロセスが開発される製品の品質を左右するからである。特に、製造工程のないソフトウェアでは、設計が品質のほぼ全てを決定づけてしまう。そのため、最初から良い品質のソフトウェアを設計できなければ、顧客の要求を満足できる製品の開発は不可能である。機械であれば、改良によって品質を向上させることが可能であるが、ソフトウェアにはそれができない。このことを理解した上で、良いソフトウェア開発プロセスとは、どのようなプロセスを言うかを理解する。プロセスを実施するのは、人間であり、プロセスではない。いかに理想的なプロセスを考えても、実施する人間が、プロセスを確実に実施できなければ、絵にかいたモチである。

6. ソフトウェア開発プロセス評価の必要性と基本的な考え方

ISO9000 の品質マネジメントと同じように、自動車の組み込みソフトウェアの良さも、それを開発したプロセスをよく分析することで、事前に評価できる。この経験則を活かして、米国防総省は、ソフトウェア開発を委託する業者を選定するために、各納入業者がどのようなプロセスでソフトウェアを開発しているかを調査し、納入業者を格付けする方法を開発した。その方法は、一般的に CMM と呼ばれている。その後、CMM は、より一般化された CMMI として再定義されたが、その DNA は、現在のほとんどすべてのソフトウェア開発プロセス評価法の基礎となっている。その CMM の基本的な考え方を理解し、また「なぜ、プロセスの継続的な改善」が重要なのかを理解する。

7. ISO/IEC15504 の考え方と歴史

従来 SPICE と呼ばれていたソフトウェア開発プロセス評価の考え方が、国際規格として提案され、標準化された。原案は、ドイツにおける研究成果としてまとめられた考え方であるが、CMM の拡張とも言えるものである。この枠組みを応用して、英国では PPA という方法が BSI によって国内規格化された。この ISO/IEC15504 には、ソフトウェアプロセスに関するもう一つの規格、ISO/IEC12207 が参照されている。これらの標準規格の内容を理解し、現場でのプロセスを記述するための基礎知識を得る。

8. ソフトウェアの非決定性と機能安全

市民の生命と財産に大きな影響を与えるかもしれないソフトウェアをミッションクリティカルなソフトウェアと呼ぶ。スペースシャトルのソフトウェアや、原子力発電所の炉心制御を実行するコンピュータのソフトウェア、鉄道や道路に設置された信号機を制御するソフトウェア、無人の電車を制御するソフトウェアなどは、みなミッションクリティカルなソフトウェアの例である。もちろん、銀行で稼働している勘定系オンラインシステムのソフトウェアも同様である。これらのソフトウェアを誤りなく開発することは不可能である。従って、誤りが残存することを仮定して、それでも人間の生命と財産に多大な悪影響を及ぼすリスクを最小化するような機能を実装することが、求められる。そのように安全を機能面で保証し、ソフトウェアの非機能的な品質である安全性を保証するかが問題となっている。このような背景から、製品の機能安全を担保するための国際規格が検討された。そのような背景を理解したうえで、自動車に組み込まれる部品等の組み込みソフトウェアでは、機能安全をどう考え、対応すべきかについて考えるための基礎知識を得る。

9. FTA、FMEA、形式手法

ソフトウェアだけでなく、システムの安全性を科学的に分析する方法として、FTA と FMEA が開発され、応用されている。また、近年ではヨーロッパを中心に形式手法(数学的方法)の適

用研究が進んでいる。これらの方法は、国際規格においても推奨されており、機能安全を設計・評価する上で、最も基本的な方法になる。従来の応用では、自動車の品質改善のために FMEA が応用され、原子力発電所の炉心制御システムの設計評価に FTA が応用されるなど、これらの手法はばらばらに適用されてきた。しかし、自動車関連分野においては ISO26262 によって、今後は、これらの方法を適確に適用して、自動車に搭載されるコンピュータを制御するソフトウェアの機能安全設計評価を実施することが求められる。従って、これらの方法の基本的な考え方と、どのように応用すべきかについて、正確な知識を持つことが重要になる。

10. ソフトウェアのテストとは

ソフトウェアのテストと、電子回路や機械部品のテストは、全く異なっている。それは、ソフトウェアのテストでは、繰り返しは全く無意味であることに起因する。電子部品や機械部品と異なり、ソフトウェアは、おなじことをすれば、同じ結果しか出てこない。これは、ソフトウェアには、摩耗のような経時変化(劣化)がないためである。ソフトウェアが故障する理由は、単に設計が誤っている部分が実行されたからにすぎない。それは、別の言い方をすれば、テストで漏れていたためである。ソフトウェアの機能の組み合わせは、簡単なものでも 100 万のオーダーになる。その組み合わせ全部を、開発期間内にテストすることは、現実的に無理である。しかし、テストをしなければリスクを背負うことになる。この問題を解決するために開発されたのが、ソフトウェアのテスト理論である。そのようなテスト理論の基礎を学ぶ。

11. 組み込みソフトウェアのテストはどう設計するのか

全ての組み合わせをテストするためには、ほぼ無限の時間が必要になる。スペースシャトルでは、極めてまれなことではあっても、開発から 40 年後に設計のミスが発見される。エアバス A300 で開発されたソフトウェアのほとんどは、A320 にも応用されている。そして、今でも問題は報告されている。このように、ソフトウェアの欠陥は、無限に存在するかのように見える。しかし、欠陥の数は有限であり、実際には全ての組み合わせをテストしなくても、誤りが存在しないことを確認できる。そのためには、入念に考えられたテスト設計が重要になる。そのようなテストの設計法と、テストのプロセスについて学ぶ。

12. 組み込みソフトウェアの信頼性評価をどうやるか

一般のソフトウェアも、スペースシャトルの組み込みソフトウェアも、エアバスの組み込みソフトウェアもどの程度の故障リスクがあるかを評価して、開発の完了を決定している。その信頼性評価に応用される理論は、機械部品や電子部品の信頼性評価に応用されている信頼性理論とは大きく違ったものである。ここでも、ソフトウェアは、同じ入力に対しては、同じ出力しか出さない性質が影響している。また、ソフトウェアの場合は、故障が起きるとその原因を究明して、ソフトウェアを修正し、二度と同じ問題が発生しないようにすることができる。すなわち、修正したソフトウェアでは、同じ問題は出ないため、故障の時間間隔が段々と長くなる性質がある。そのような性質を持ったソフトウェアの信頼性とはどのようなことかを理解し、その評価法の基礎を学ぶ。

7	制御システム開発事例
担当講師	広島県立総合技術研究所 西部工業技術センター 生産技術アカデミー 製品設計研究部 副部長 大賀 誠
概要	本研修事業で学ぶモデルベース開発の技法を活用した開発事例を紹介する。開発した題材は台車型倒立振り子システムで、一般的なモデルベース開発の工程を上流から順を追って解説することにより、本手法を活用した具体的な開発手順の理解を図る。
使用教材	資料を配布
講義の流れ ポイント	<p>MATLAB/Simulink を活用したモデルベース開発を、台車型倒立振り子システムを例に紹介する。今回の開発工程を図に示す。図中の番号順に設計から実装、検証と工程を進め解説を行う。</p> <p>①制御対象モデリング 制御対象となる台車型倒立振り子のモデリングを行う。物理量をパラメータとして設定し、ラグランジュの運動方程式を解くことで状態方程式を得る。</p> <p>②コントローラモデリング（制御系設計） 制御設計によりコントローラをモデリングする。現代制御理論を採用し、多入力制御系を扱うことが可能な状態フィードバック制御で設計する。</p> <p>③モデルシミュレーション 設計した制御対象とコントローラのモデルを使用してモデルシミュレーションを行う。これにより倒立振り子制御が実現可能かを確認する。</p> <p>④リアルタイムシミュレーション コントローラのモデルで実機を制御することで、実時間での制御設計検証を行う。一般的にこの工程をラピッド・コントロール・プロトタイプ（RCP）と呼ぶ。</p> <p>⑤自動コード生成・実装 RCP により検証が終了したコントローラモデルから、C 言語のプログラムコードを自動生成し、マイクロコンピュータ（MPU）へ実装する。</p> <p>⑥実機検証 実装が完了した MPU を用いて倒立振り子制御の実機検証を行う。</p> <p>図 モデルベース開発の工程</p>