

テストィング（基礎）

平成27年度シラバス

2015年1月9日

国立情報学研究所

トップエスイープロジェクト

代表者 本位田 真一

1. 科目名

テストイング(基礎)

2. 担当者

鷺見 毅

3. 本科目の目的

本科目では、ネットワーク家電の制御ソフトウェアを題材とした産業ソフトウェアのテストイング問題を扱う。近年ますます大規模化・複雑化するソフトウェアの設計において、その動作の正しさを保証することがより困難になっており、プログラムの動作の自動的、または半自動的なテストイングの支援を行う各種ツールを使用し、実際のシステム開発に適用する方法を習得する必要がある。しかし、これらのテストイングツールを使いこなすには多くの難しさがあり、普及への障害となっている。そこで本科目では、現実的なプログラムのテストイングを通して、ソフトウェア・テストイングの難しさを体感した上で、テストイングツールの適用における難しさを解決するためのテストイングプロセス、およびノウハウの体得により、組込みシステムからネットワークまで対応可能な、ソフトウェア・テストイングの技術を習得する。また、グループ討議を通してソフトウェア・テストイングの課題を正しく認識し、テストイングプロセスの実用的ノウハウを体得できる効果が期待できる。

4. 本科目のオリジナリティ

テストイングツールを使用したソフトウェア・テストイングについては、従来多くの同様な講座・科目が開講されている。しかしそれらの講座・科目にはさまざまな問題点があったため、受講生が習得した内容を速やかに開発現場で適用するための障害となっていた。本科目では、それらの問題点を解消し、本科目受講後テストイングツールを速やかに開発現場で適用できるように配慮している。表 1 に、既存の講座・講義の問題点と、本科目における解を示す。

表 1 既存の講座・講義の問題点と、本科目における解

既存の講座・講義の問題点	本科目における解
テストの概念と基本的な技法の習得を目的とした簡単な例を扱うのみで、開発現場での適用が難しい	基本的な例題から、複雑で現実的な問題に順に取り組むことによって、テストイングの基礎的素養を養うため、現実的な開発プロセスへのシームレスな組入れが可能

5. 本科目で扱う難しさ

近年、家電の制御ソフトウェアは大規模化、複雑化が進んでいる。特に、複数の家電機器がネットワークに接続され、相互に連携して動作することで 1 つの機能を実現する、ネットワーク家電市場が急速に立ち上がりつつある。従来の家電機器の制御ソフトウェアは単体で動作するように設計されていたのに対して、ネットワーク家電では、他の機器との連携動作のためのプロトコルなどが必要になるため、制御ソフトウェアが非常に複雑になる。さらに、ホームネットワークでは、動的に変更される接続相手の識別や、不安定なネットワーク環境などを考慮する必要がある。その一方で、大規模・複雑化、短納期化、頻繁なる仕様変更対応など、開発体制の環境も厳しさを増している。このような状況においては、品質を確保するための網羅性と効率性を両立したテスト手法を用い、かつ効果的であることが求められている。また、既存開発や外注委託等の調達モジュールなど、外部モジュールとの組合せで製品開発を行い、品質を確保しなければならない状況にある。

5. 本科目で習得する技術

将来のネットワーク家電では、接続する機器や機器間で交換する情報の数・種類が従来型の家電機器よりも大幅に増加するため、その振る舞いの設計が大規模化、複雑化し、従来のアドホックなテストでは、プログラムの正しさを効果的にテストすることが困難になる。そこで近年では、テストケース作成やテスト計画における体系化されたプロセスや、単体テストツールやテスト管理ツールなどテスト作業を自動的、または半自動的にサポートする各種ツールの利用が進みつつある。さらには、テスト作業を中心にソフトウェア開発プロセス全体を再構築した、テスト駆動開発プロセスも広く知られてきている。

しかし、大規模ソフトウェア開発に、これらのプロセスやツールに代表される、ソフトウェア・テスト技術を利用する場合には、各種技術の背景理論を理解した上で、様々なノウハウを駆使しなければ、効率的かつ効果的に活用することができない。たとえば、テストケース作成プロセスにおいては、テストケースを構成するデータ構造を熟知する必要があり、またテスト駆動開発プロセスにおいては、従来の開発プロセスの常識から離れる必要すらある。

本科目では、ソフトウェア・テストの基本として、ブラックボックステストやホワイトボックステストなどの概念、テストケース作成や単体テストなどのプロセスや技法を習得する。さらに、実問題への適用を可能にするためのノウハウとして、直交表に基づくテスト効率化手法などを習得する。そして、テスト駆動開発プロセスの習得により、ソフトウェア・テスト技術をベースとしたソフトウェア開発プロセスを習得する。

6. 前提知識

本科目の受講生は、以下の項目を習得済みであることが望ましい。

- ・ Java プログラミング

7. 講義計画

- ・ 概要

第1回：テストの基礎（プロセス・技法）、テストの課題

第2回：テストツール

第3回：テストツール

第4回：テスト駆動開発（概念とポイント）

第5回：ブラックボックステスト（機能テスト）

第6回：ブラックボックステスト（機能テスト）

第7回：直交表（テストの効率化）

第8回：直交表（テストの効率化）

第9回：直交表（応用テクニック）

第10回：直交表（応用テクニック）

第11回：ホワイトボックステスト（パスカバレッジ）

第12回：テスト駆動開発

第13回：テストの管理と評価・改善

第14回：テストの管理と評価・改善

第15回：最終演習、まとめ

- ・ 詳細

(1) テストの基礎（プロセス・技法）、テストの課題

テストの基礎(座学)

- テストの基本概念
- テスティングプロセスと技法の概要

テストの課題（グループ討議）

- 開発現場におけるテストの課題の討論

(2) テストツール

- Eclipse + JUnit を使ったテストの実施

(3) テスト駆動開発

- テスト駆動開発の基礎と実際

(4) ブラックボックステスト（機能テスト）

- ブラックボックステストとは？
- テストケース作成の基本
- デシジョンテーブルの使用

- ネットワーク家電例題を用いた演習
- ブラックボックステストとテストツールでの実装の関係

(5) 直交表

- 直交表とは？
- 直交表を用いたテスト効率化
- 直交表作成実習
- 直交表を用いたテスト効率化実習

(6) ホワイトボックステスト（パスカバレッジ）

- ホワイトボックステストとは？
- カバレッジとは？
- Eclipse + JUnit + djUnit でパスカバレッジ計測
- パスカバレッジを使ったテスト管理

(7) テスト駆動開発(2)

- 従来のテスト手法・開発手法との違い
- テスト駆動開発で機能拡張をしてみよう（演習）

(8) テストの管理と評価・改善

テスト管理の基礎

- テスト管理とは？
- データの一元管理とテストの状況の見える化
- バグ管理とテスト管理の関係

テスト管理の実際

- テスト管理ツールを使ったテストの管理（演習）

(9) 最終演習・まとめ

- テスト駆動開発での機能拡張（演習）
- テスト技術の使えるところ、使えないところ（課題）
- まとめ

8. 教育効果

本科目を受講することにより、実際のシステム開発、特に組込みソフトウェアの開発に適用可能な、ソフトウェア・テスト技術を習得できる。その結果、開発現場において、テストツールを活用することにより、信頼性の高いシステムを、効率的に開発することができるようになる。

9. 使用ツール

JUnit : 単体テストツール

- ・ 使用する上での難しさ
 - テスト用コードの記述が難しい
- ・ 使用上必要なノウハウ
 - テスト用コード記述ノウハウ
 - ◇ テストケースの表現
 - ◇ アサーションの表現
- ・ 選択理由、実用性
 - **Java** プログラムの単体テストツールとして広く使用されている
 - 特にテスト駆動開発には必須

10. 実験及び演習

2～3 名程度の少人数で構成されたグループ単位で議論を行い、議論を通してソフトウェア・テスト技術の理解を深める。講義内でソフトウェア・テストのプログラミングを行い、難しさを実際に体感しながらソフトウェア・テストプロセスを体得する。

11. 評価

課題レポート、講義・演習の理解度、出席日数等を総合して評価する。

12.教科書/参考書

- Rick Craig (著), Stefan P Jaskiel (著), 成田 光彰 (翻訳), 宗 雅彦 (翻訳), 「体系的ソフトウェアテスト入門」, 日経 BP 出版センター, 2004.
テストの分析、設計・実装、実施、マネジメント、改善まで幅広く扱っており、この科目に最適である。
- Rex Black (著), テスト技術者交流会 (翻訳), 「基本から学ぶテストプロセス管理—コンピュータシステムのテストを成功させるために」、日経 BP 社、2004
テスト管理手法について体系的に解説しており、この科目に最適である。