



# いまどきのテスト教材

小菅 貴彦

KOSUGE, Takahiko  
kosuge@jec.ac.jp

情報系 主任研究員  
(学)電子学園 日本電子専門学校



学校法人 電子学園

日本電子専門学校

# 本日の内容

- 自己紹介
- ソフトウェアテストデザイン科のカリキュラム
  - 学科の設立趣旨説明
  - カリキュラムの設計指針
- 実際に使用する教材
- 模擬講義
  - 伝統的な「三角形判定プログラム」を使って
  - ぜひ、ご参加ください
- おわりに





# 自己紹介

- 小菅貴彦 (こすげ たかひこ)
  - 千葉大学 理学修士 (専攻: 超伝導)
  - 現在 日本電子専門学校 情報系主任研究員
- 経歴
  - こてこての実装屋 (組込み、言語、Webアプリなど)
- これまで公開したソフトウェア
  - USB-KNOPPIX、GreenBarbarian
  - openCASL、algolAlpha



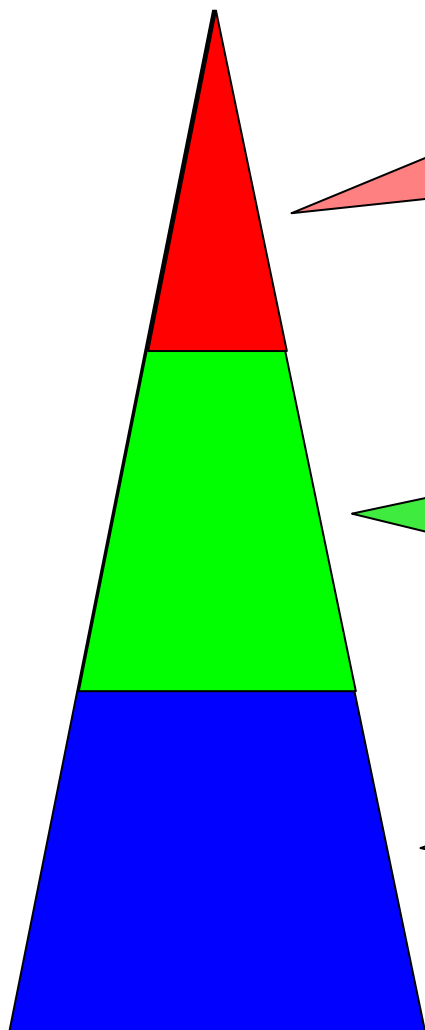
# ソフトウェアテストデザイン科

- 今年4月(2008/04)より開講します
- テスト技術者養成を主たる目標にした、(たぶん)日本で初めての学科です
- コンピュータの素人を2年間でソフトウェア品質管理、テストのエキスパートに育てたいなあ
- 理論に偏りがちな教育ではなく、実際に様々なソフトウェアに触れながら、実習・演習を中心とした、カリキュラムを組んでいます





# どのような技術者を養成するのか



## テストプランナー

ソフトウェアの品質管理に責任を持ち、テスト計画を立案する上級技術者。深い知識と経験が必要。教育機関で育てることは困難

## テストデザイナー

テスト計画に基づき、テストセットを作成する中級技術者。知識と経験が必要

**ここが目標**

## テスター

テスターテストを実際に実施する。極端な話アルバイトやパートでも一定の教育を受ければ大丈夫



# テストエンジニアに必要なスキル

- テスター
  - さまざまなソフトウェアテストを実施することができる
- テストデザイナー
  - テストケースを発見し、テストセットを作ることができる
  - 必要なテスト技法を選択することができる
  - テストに必要な環境を整えることができる
  - 統計的な手法でテスト結果を分析することができる
- テストプランナー
  - テストの実施要項を決めることができる

# 何を教えるか

- 業界の専門用語
  - 技術者として会話に参加する最低の条件
- ソフトウェアテストの概論
  - 開発プロセス、レビューの方法など
- テストおよびテスト設計の技法
  - テストの種類と特徴、テストのマネジメントなど
- その他
  - トラブル事例、現場のテスト技術者による講演
  - 一般教養的統計学



# どう教えるか


- 経験を積むために総時間の1／3を実習・演習に
  - 実際にテストを行う、テスト計画を立てる
- 客観的な知識量の目標として資格の取得
  - JSTQB など
- オープンソースプロダクツの利用
  - Mozilla Suits、OpenOffice、NetCommons(CMS) など
- 最終学年(2年)時にインターンシップを
  - まだ先ですが、受け入れ企業さん、募集しています！





# どんな学生を育てるか

- 卒業生のスキル
  - テスターとしては即戦力
  - 経験を積めばテストデザイナーに
  - 将来のテストプランナー
- 素性の良い学生
  - 特定のツールに依存しない
  - 本学では特定の色は付けませんので、採用した企業さんで好きな色に染めてください。ただし染め易さは保証します



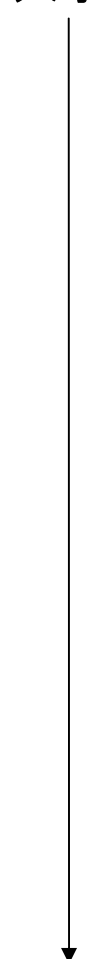
# 入学者予定者

- 留学生
  - 中国・韓国からの留学生が中心
- 女性
  - 既卒で、現状に不満があり、スキルアップを狙う？
- 先進的な学生
  - すでにプログラミングは極めた？学生

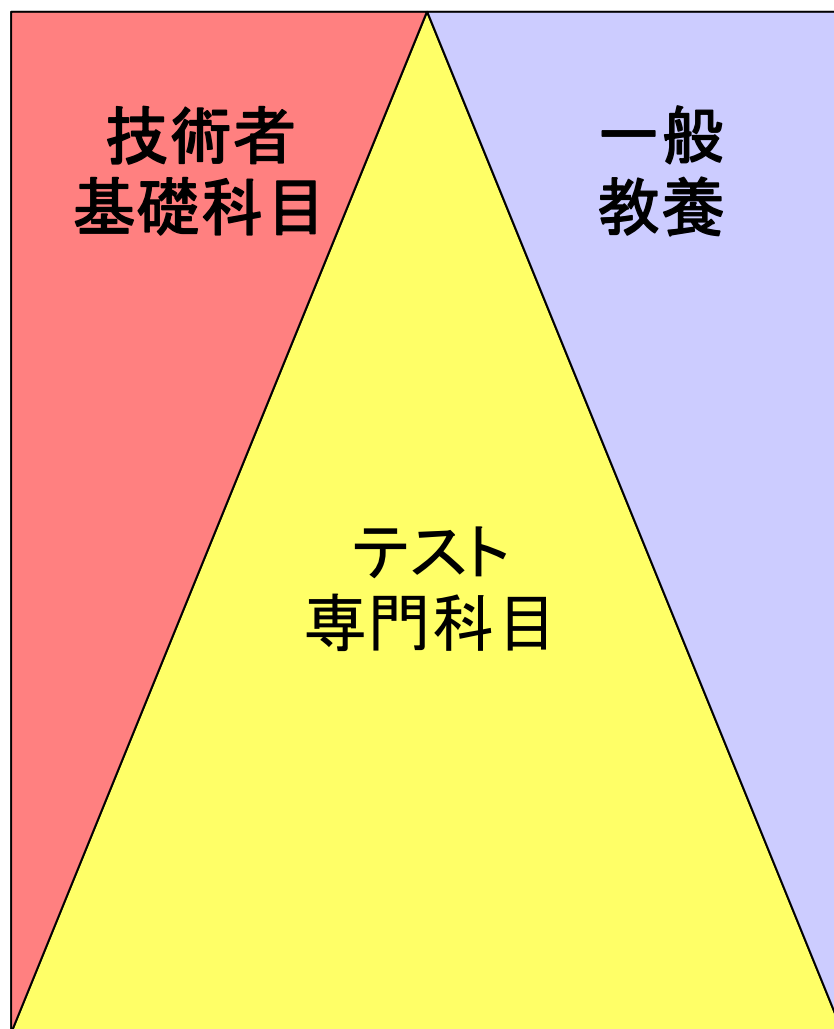


# カリキュラムの方針

入学



卒業



- 一般教養
  - リテラシーや発想法
- 技術者基礎科目
  - 資格試験の学習を通じて技術者としての素養を身につける
- テスト専門科目
  - 実習や演習が中心
  - 2年次にはインターンシップも



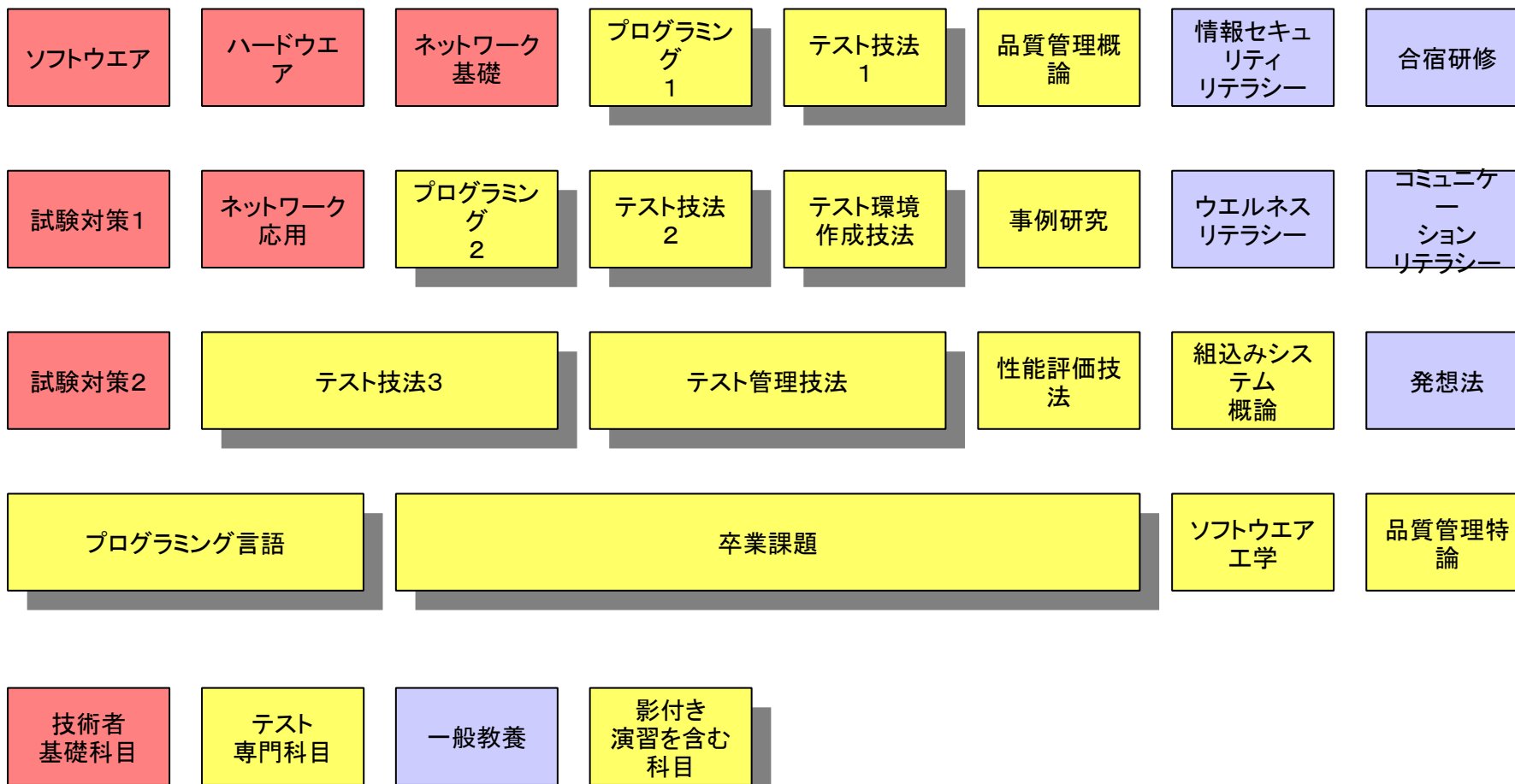


# カリキュラム構成(概略)

入学



卒業





# カリキュラム構成(詳細) 1年前期

期	分野	科目名	内容	講義	実習	期ごと		
						講義	実習	合計
1前	A	ソフトウェア	情報処理技術者試験の学習を通じて、ソフトウェアの基礎を学習する	3	0	18	7	25
	A	ハードウェア	情報処理技術者試験の学習を通じて、ハードウェアの基礎を学習する	4	0			
	C	ネットワーク基礎	ネットワーク(TCP/IP)の基礎を学習し、実習ではネットワークの設定やコマンドを学習する	2	1			
	B	プログラミング1	テスト作業の自動化に必要な制御構造を簡単なシェルプログラムを通じて学習する	3	3			
	D	テスト技法1	既存のソフトウェアを使って不具合を発見すると同時に、バグ再現化手法、バグレポートの書き方などを学習する	2	3			
	D	品質管理概論	ソフトウェアの品質について基礎的な学習する	2	0			
	E	情報セキュリティ	コースウェアを使用してコンピュータセキュリティについて学習する	0.5	0			
	E	那須特別研修	那須研修所でスポーツなどをおこなう	0.5	0			
	E	H.R.		1	0			





# カリキュラム構成(詳細) 1年後期

1後	A	試験対策1	基本情報技術者試験対策	2	0	15	9	24
	C	ネットワーク応用	比較的多く利用されるインターネット上のプロトコルを学習すると同時に、実習を通じてネットワーク操作コマンドの利用法を学習する	2	2			
	B	プログラミング2	プログラムを作成するのではなく、オープンソースのプログラムを利用して、テストに必要なプログラムのビルドを学習する	3	2			
	D	テスト技法2	オープンソースのプログラムをテスト自動化ツールを利用して、テストを作成して実施する	3	3			
	D	テスト環境作成技法	さまざまなソフトウェアをインストールして、テストのための環境を整える方法を学習する	0	2			
	D	事例研究	事例学習として過去のソフトウェア事故を学習し、その原因などについての知識を得る	2	0			
	E	ウェルネスリテラシ	社会人として必要な健康管理方法などに付いて学習する	1	0			
	E	コミュニケーションリテラシ	社会人として必要なプレゼンテーション技法などを学習する	1	0			
	E	H. R.		1	0			





# カリキュラム構成(詳細) 1年後期

2前	A	試験対策2	テスト技術者資格試験対策	2	0	15	9	24
	D	テスト管理技法	ソフトウェアの開発プロセスを解説し、そのなかでテスト計画を立て、ソフトウェアの品質管理をおこなう技法を学習する	3	0			
	D	テスト技法3	オープンソースのプログラムを利用して、単体テストから総合テストまでを一貫して実施する	2	6			
	D	性能評価技法	オープンソースのデータベースを利用して、ソフトウェアの性能試験を実施する	2	3			
	A	発想法	KJ法、マインドマップなど、レビューを円滑に開催し、アイデアをまとめる方法を学習する	2	0			
	A	組み込みシステム概論	先進的にテスト導入を行っている組み込みシステムの概略を学習する	3	0			
	E	H.R.		1	0			





# カリキュラム構成（詳細） 1年後期

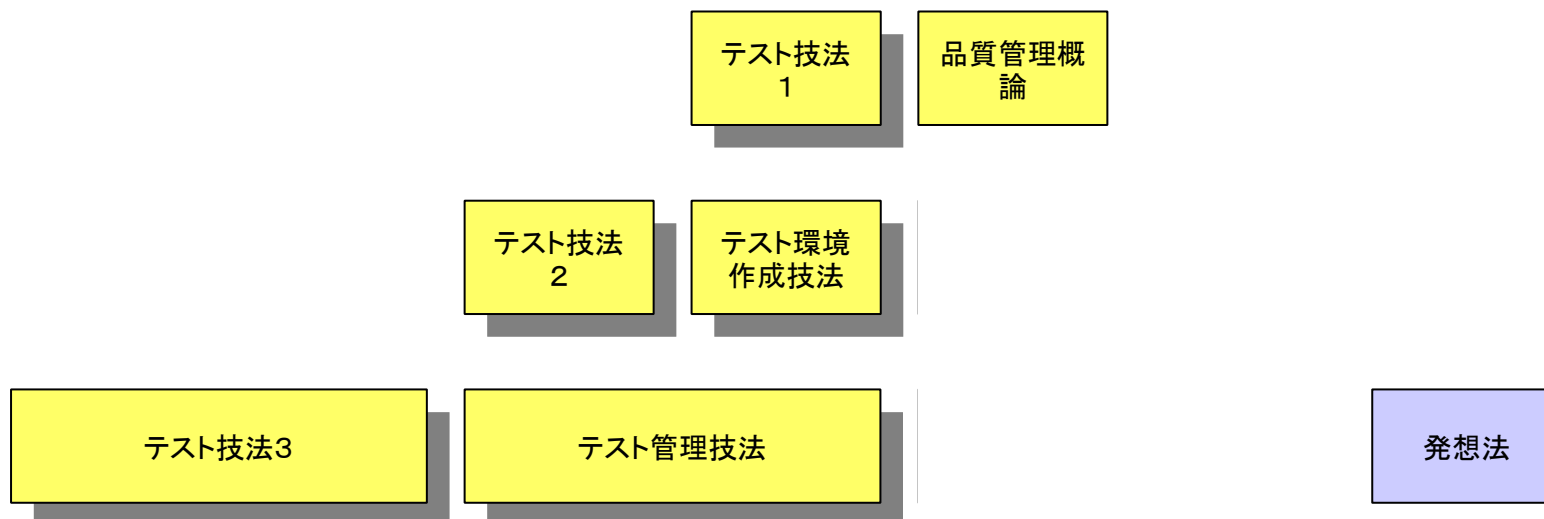
2後	D	卒業課題	各々の選択したオープンソースのソフトウェアを、そのオープンソースのコミュニティと連携して、品質管理のためのテストを実施する	0	10	11	13	24
	D	品質管理特論	実際にソフトウェア開発の現場にいる技術者の講演を聞く	3	0			
	A	ソフトウェア工学	作業見積りなどのソフトウェア工学を学習する	3	0			
	B	プログラミング言語	C,Javaなどのプログラミング言語の概略を学習する	4	3			
	E	H.R.		1	0			







# 社会人向けのカリキュラム



上の科目を中心に教材を公開します





# シラバスの例(テスト技法1)

講義週	内容	内容
1	教材の概要	教材とする「三角形判定プログラム」の概要の説明します
2	出たところ勝負テスト	何も考えずにデータを投入し、不具合を探します
3	ブラックボックステスト	仕様書を読みプログラムの動作を理解します
4		同値クラスを見つけてテストケースを作ります
5		境界値分析を行いテストケースを作ります
6		デシジョンテーブルの利用方法を学びます
7		因果グラフの利用方法を学びます
8		直交表の意味と利用法を学びます
9		ペア構成テストを実施します
10	ホワイトボックステスト	設計仕様書を読み教材の内部構造を学びます
11		制御フローテストを学びます
12		データフローテストを学びます
13		データフローテストを学びます
14		
15	まとめ	全体をまとめます



# 教材の検討

- バグのコントロールされたソフトウェアが必要
  - テスト教材のために作成したソフトウェア
  - これまで本学で作成してきたソフトウェア
  - オープンソースなソフトウェア
  - プロプライエタリなソフトウェア
- 幸いなことに本学には、本学が独自に作成したオープンソースプログラムが多数存在している
  - これらのバグを分析してテスト用教材として利用する



# 教材のレベル

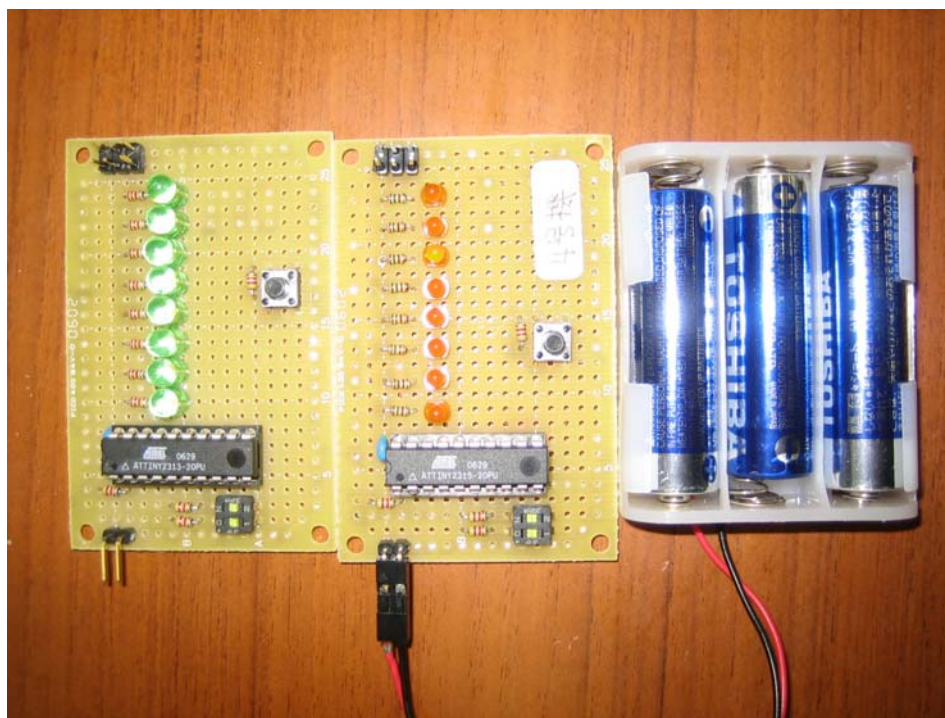
レベル	代表的なテスト	内容
入門		非プログラム 間違い探しなどクイズレベル 体験学習などで使用
初級	ブラックボックステスト	仕様書レベルでの検討を要するバグ テストセットを与えてテスト実施 仕様書からテストセットを作る
中級	ホワイトボックステスト	制御フローを考慮した検討を要するバグ プログラムの簡単な知識が必要 テストセットを作る
上級	結合・負荷テスト	デッドロックなどの輻輳条件の検討を要するバグ 背後にDBのあるWebアプリケーションなど 過去のトラブル事例などの知識が必要 テストセットを作る

# オープンソースソフトウェアの教材

- Mozilla Suits
  - Firebird、Thunderbird、Sunbird
- OpenOffice
  - Writer、Calc、Impress、Draw
- exchecker
  - Web上で動作する実習・演習管理システムです
- 不具合の混入している過去のバージョンを保存してあります

# 組み込みの教材

- コンピュータ教育開発センターのOpen School Platform プロジェクトで作成したワンボードマイコンの廉価版を作成しています





# 教材の例(三角形判定プログラム)

- テスト教材として伝統的な物です
- 3つの実数の数値を入力します
  - 入力された数値をそれぞれ三角形の辺の長さと考えて、「正三角形」、「直角二等辺三角形」、「二等辺三角形」、「普通の三角形」、「三角形でない」を判定します
- 実際に私が作成したものに、(ほんとは)絶妙の不具合が混入していました
  - ブラックボックステストで見つけることができます
- それを修正した結果、別な不具合が発生しました
  - ホワイトボックステストの方が見つけやすいものです





# さて！ 模擬講義です

PCをお持ちの方はぜひご参加ください

詳細は、当日配布する資料をご参照ください

