課題1

以下のようなシステムを開発したい。
適当なソフトウェア開発プロセスについて理由をつけて
説明しなさい。

(a) 自動車に組み込みのABSのソフトウェア

解説ありがとう

- ABSの原理は次のとうりである。

- 1. ホイールセンサー（W.S.）が各車輪の回転速度を検出し、エレクトロニックコントローラー（E.C.U.）へ信号を送ります。
- 2. E.C.U.は、W.S.から伝えられた車輪速度がプログラムに入力された一定の減速パターンより低減したら、ロック傾向に入ったと判断し、ハイドロリックユニット（H.U.）に指令を送ります。
- 3. H.U.は、E.C.U.からの指示に従い、バルブの開閉作動をして、ブレーキ液圧を減圧、保持、増圧し、適正なスリップ範囲になるようにブレーキを制御します。

(a) の回答分布

- Waterfall Model　23
- Evolutionary/Prototyping　6
- Spiral Model　1
- Formal Trans.　3
- その他
  - Waterfall か Prototyping　1
  - Waterfall か Formal　1
  - 問題の意味がわかっていない　1

Waterfall Model 支持派

- 複雑ではなく、全体がよく見えている。
- 仕様がよく理解されている。
- 安全性が求められるので可視性が必要。(？)
- 後戻りが生じる恐れが少ない
- 頑健性、保守性、信頼性(？)
- なるべく軽量動作する必要(？)
- 人の命に関わるものであるため、確実に動作しなければならない(？)
- 大規模ではなく、発展させていくという形式をとる必要がない(？)

Evolutionary 支持派

- 自動車を用いた実験は莫大なコストがかかるため、ソフトウェア開発段階で出来るだけ完璧なものを作り上げ、無駄な実験を減らす。(？)
- 小規模な部類に属する
- 実験を重ね、問題が発生したらその原因を見極め、それを改善することが重要である。
- ABSが既知ではない、もしくは小さいシステムであればprototypingで開発

Spiral 支持派

- 人命に関わるため、特に完成したシステムに問題があってはならないような種類の製品なので、各段階ごとに何度も試作品を作って、要求仕様や使用条件を確認して、開発に当たる必要がある。

Formal Translation 派
（間違いです）

- 数学的モデルを実装する形で開発が行われる。
- 車輪が滑らない時に自動車の速度（v）と角速度（w）と車輪の半径（r）の関係はv =w*r （式1）と表せる。車輪が滑ると（式1）が成立しなくなる。（式1）が急ブレーキ時に成立するように、ABSのソフトウェアは車輪の角速度（w）を制御するソフトウェアである。だから、このようなソフトウェア開発プロセスは Formal Transformation（形式的変換）型である。つまり、数学系モデルを実用化するソフトウェアである。

(b)バーチャルリアリティを用いた遠隔会議システムのソフトウェア

- Evolutionary/Prototyping　20
- Spiral　5
- Prototyping 中心で他のモデルも併用　2
- Waterfall　3
- 答えになっていない　2

Evolutionary 支持派

- ユーザの使いやすさが最も重要
- ネットワークの負荷・参加人数等により予測が困難な不具合が生じる可能性
- バーチャルリアリティは、まだ研究が進んでいる比較的新しい分野である
- 過去に蓄積されたサンプルがあるわけではない
- 実際に試作品をユーザに試用してもらって、機能・操作性などを早期に検証してもらい要求仕様を確認する必要がある。
- 改良を行うことが多いと予想され、システムを前もって完全に仕様化することは不可能であると考えられる

Spiral 派

- 様々な要素が関わってくるため、これらの要素を、少しずつ開発、実装してゆく。
- とりあえず形になったところで市場に出し，あとは現場から出た要求をもとに改良していけば良い。
- ユーザの使いやすさが最優先。さらに、開発後の保守や拡張が容易にできることが大切。
- user interface等最初の段階では見通しがつかないことが多そう
- 仮想空間では画像を扱うため、実際に作成したものが顧客が期待したとおりではなかったり、他によいアイデアを考え付いたりして次々に改良されることが考えられる。
- 利用者の反応を見ながら開発が進められ、効率よく開発できる

Evolutionary+ α

- ある程度以上、規模が大きいシステムであればprototypingの方法によってその一部を作成し、全体としてはWaterfall modelを用いるなどの対応を取らなければならなくなる
- (Evolutionary developmentであるが、) もし少しばかりの実績があり、プロセスについて少し見えるのならば、Spiral model で行うと visibility 等の面でなお良い。
- 通信部分はある程度仕様をはっきりさせることが出来る場合には通信部分はウォーターフォールモデルによって堅牢に開発し、ユーザに近い部分だけをプロトタイピングで作成することも良い。

Waterfall 派

- 何度でも実験をする事が出来、また、ユーザを交えて実験を重ねていくうちに、ユーザもシステムに対する要求が増えていくものと思われる。これらに対応出来る様に。(？)
- ユーザの要求は「リアルタイムで会議ができること」とはっきりしている。(？)
- 大規模なシステムなので。(？)

(c)大学の経理システムで、以前からあるシステムをリプレースする場合。ハードウェアと**OS**に大きな変更はない

- Reusing　14
- Waterfall　11
- Reusing and/or Waterfall　2
- 場合により使い分ける　1
- Formal Trans.　2
- Evolutionary　2
- 答えになっていない　3

Formal Trans. 派（間違い）

- 広義のバージョンアップと考えてよく、それほど難しい作業が必要なさそうなのでトランスフォーメイショナルモデルが適していると言える。
- 全てを同時にリプレースする必要はなく，徐々に部分ごとを置き換えていけば良い。迅速にことを行うため大人数が一度に関わる必要はないが，後々のために可視性は優れていなくてはならない。

Evolutionary 派

- 開発システムのバージョンアップと位置付け、発展系開発モデルを用いるのがよい。(？)
- 以前のシステムと互換性のあるものでなければならない。また大学の経理システムの場合には、新たなシステムに移行するために新たに使い方を学ばなければならないという課題がある。したがって理解容易性も重要である。またいつかは再びリニューアルされるシステムである。以上より、Evolutionary DevelopmentのThrow-away Prototypingによる開発プロセスにするべきである。(？)

課題2
国家資格にするべきかどうか?

- 賛成　14　(うち、問題点を指摘が2)
- 反対　16
- 賛成だが困難　1
- 倫理面のみ試験　1
- 管理者のみ試験　1
- 一部のソフトのみ試験　1

賛成意見の例

- 責任が重大
- 勉強の動機づけになり、レベルが向上する
- 共通知識をもたせる
- 資格を与えることより剥奪することに意味がある
- 品質の信頼性
- 倫理観が問われている

反対意見の例

- 自由な発想やベンチャーを阻害
- Linux (オープンソース)やフリーウェアの開発の例
- 人材が逃げる
- 経験の方が重要
- 芸術性がある
- 医者や弁護士ほどの重大責任はない
- 進歩が急で、試験内容がすぐ陳腐化する。また過去にとった資格が意味をなさない。
- 誰が開発したのかを検証できない
- 試験範囲が広範囲すぎる
- 試験回答能力と開発力は別
- きまりごとが少ない

課題3

- 「良いプログラマが必ずしも良いソフトウェア開発管理者になるとは限らない。その理由を説明せよ。」
- これは、だいたいできてました。

管理者の仕事

- ◆提案書を書く
- ◆費用見積り
- ◆計画、スケジュール
- ◆監視とレビュー
- ◆要員確保と評価
- ◆報告作成とプレゼンテーション
- これらは、プログラマの仕事とは異なる。
- ただし、プログラマ経験は重要

課題4

- 「次の要求を、validate 可能なように書き換えなさい。」
- 問題の意味を理解していない人がいました。
- Validate 可能とは
  - 曖昧性がなく、誰が読んでも同じ意味にとれる
  - あとでトラブルや訴訟にならない
  - つまり、要求が満たされているかどうか客観的に判断できる。

(a) 解説と解答例

- そのソフトウェアシステムは、最大負荷がかかったときでも十分な処理能力を示さねばならない。
  - そのソフトウェアシステムは、同時アクセスが100ユーザを超えても、ユーザへのレスポンスに0.1sec以上要してはいけない。
  - その電子掲示板システムは、一日10万ヒットの際の回線混雑時の最大ヒット数毎秒400ヒットでのテストにおいても、ダウンせず、5秒以内にユーザーに対して応答する能力を保持する

(a) 誤答例

- そのソフトウェアシステムは、ネットワークビジー率が90%を越えたときにも、必ず管理者の接続要求には素早く応えなければならない。
- そのソフトウェアシステムは、すべてのプロセスを動作させても、ある値以上の容量のメモリが空いており、スループットも基準値以内になっている。
- そのソフトウェアシステムは、十分多くのタスクがユーザから与えられたときでもシステムがダウンせずに正常動作しなければならない。

(b) 解説と解答例

- そのシステムのユーザインタフェースは、標準的な端末で利用可能な文字コードセットを利用する。
  - このシステムのユーザインタフェースは、Windows95以上の環境を持ったPC上で利用可能なシフトJISコードセットを利用する
  - そのシステムのユーザインタフェースは，文字コードセットISO-8859-1を利用する

誤答例

- ◆標準的な端末→端末の中で、最も使用されている端末としてある端末を想定して
- ◆そのシステムのユーザインターフェースは、一般的に使われているOS，つまりWINDOWS, Macintoshで扱うことができる文字セットを利用する。
- (b) はだいたいできてました。
- 「そのシステム」はたしかに曖昧ですが、問題の本意ではありません。

(c) 解説と解答例

- 今回の開発では、構造化プログラミング手法を用いる。
  - ◆ 今回の開発では、構造化プログラミング言語C＋＋を用い、違うモジュールのデータやメソッドは必ず違うネームスペースに置き、あらかじめ定義したインターフェースとなる関数**郡**でモジュールにアクセスする構造化プログラミング手法を用いる。
  - ◆今回の開発では、VC++のMFCを用いたクラスベースのオブジェクト指向プログラミングを用いる。
  - ◆今回の開発では、構造化プログラミング手法を用い、作成したコードはすべて、プログラムを検証するアプリケーション「＊＊＊＊」によって妥当であるとされるようにするものとする。

微妙な解答例

- 今回の開発では、ソース上で「go to」文がない。
- 今回の開発では、制御構造として例外処理以外のGoto命令を使わず、朝の朝礼の際に社歌の後にみんなでダイクストラの本を斉唱する。

誤答例

- 今回の開発では、以下の部分を、この様なクラスを作成する事により開発する。(以下、クラスの説明…)
- 今回の開発では、設計仕様の段階で、プログラムフロー図を必要とする。
- 今回の開発では、プログラミング手法としてスパイラルモデルを用いる。
- 要求ドキュメントの中には「どうやってするか」よりも「何をするか」を盛り込むべきである。よってこの文は削除