Computing Science GLASGOW

PSD3 2008-09

Introduction to Professional Software Development

Semester 1
Lecture 1

some basics about the course

- concerned with software development in the large
  - » specification & modelling
  - » management
  - » maintenance
- built around practical work
- emphasis on personal professional development

some basics about the course

- we take up where OOSE left off and …
  - » look at to multi-person projects
    - scale
    - coordination
    - communication
  - » expand and deepen the material on the development process
    - requirements
    - software modelling in UML
    - linked to Java (unit testing, refactoring, etc)

useful information

- staff
  - » Phil Gray
    - Module Coordinator
    - Management, Design
  - » Ray Welland
    - Analysis, Re-engineering, Formal Specification

more useful information

- textbooks (in order of importance)
  - » Ian Sommerville. *Software Engineering*, 8th edition. Addison-Wesley, 2004.
  - » Bennett, McRobb & Farmer. *Object-Oriented Systems Analysis and Design*. 3rd Edition. McGraw-Hill, 2006.
- both books are recommended, not required

and more …

- course website
  - » http://fims.moodle.gla.ac.uk/course/view.php?id=128
  - » includes
    - – timetable
    - – information about group exercise
    - – on-line copies of lecture notes
    - – recommended reading
    - – related links

and even more …

- Other useful information sources
  - » www.acm.org
    - Check out
      - the Digital Library
      - CACM
      - Queue
  - » The main IT vendors (IBM, Sun, MS)
    - www.alphaworks.ibm.com
  - » for academic papers
    - – http://citeseer.nj.nec.com/cs
    - – Google academic search engine

and most important…

- PSD3 is twice the credit value of other modules
- Runs over two semesters
- Assessment is
  - » 50% practical work
    - Half in each semester
    - Group and individual work contribute to total
  - » 50% exam

course aims

- introduce you to modern software development methods & techniques
- provide you with an opportunity to apply methods & techniques

course aims

- make you aware of professional, social and ethical dimensions of software development
- instil in you a professional attitude towards software development

term 1 plan

- management lectures
- analysis and design lectures
- workshops
  - » Monday afternoons
  - » tutorials
  - » supervised project work

term 1: software project management

- group organisation
- project planning
- documentation
- change management
- quality assurance
- risk analysis & management
- software process models
- CASE tools
- panel: “Software Development in the Real World”

term 1: analysis and design

- Requirements Capture
- Requirements Specification using Use Cases
- Prototyping
- Domain Modelling
- Modelling Behaviour
- Basic Software Design Patterns

a definition to remember…

Software engineering consists of *principles, methods, techniques and tools* for the

- specification
- development
- management and
- evolution

of software systems

Why does this matter?

- Lots of projects fail … or are not as successful as they should be
  - » Poor quality
    - Low reliability
    - Low usability
  - » High cost
- Look at
  - » www.csl.sri.com/users/neumann/insiderisks.html
  - » http://catless.ncl.ac.uk/Risks

What are the problems?

- Often not a programming issue
- Thus, the problem might that you are
  - » Building a system for the wrong reason
  - » Building the wrong system
  - » Building the system wrong

LAS: A Lesson For Us All

- London Ambulance Service Automated Despatch Service
- Abandoned shortly after delivery in 92
  - » Cost £43m
  - » May have caused up to 30 deaths
  - » Check out

  www.cs.ucl.ac.uk/staff/A.Finkelstein/las.html

What Was the Problem?

- Building a system for a bad reason
  - » Political imperative to decentralise & increase efficiency
  - » Assumption that computerisation was the solution
  - » Procurement faulty
  - » Lowest bid taken - no one asked “Why so cheap?”
- Building Wrong System
  - » Clients not consulted during development
  - » Future users ignored
  - » User hostility toward system
- Building system wrong
  - » Undocumented code changes
  - » Not much testing
  - » Disorganised deployment
  - » Standards ignored

Consider the BCS Code of Conduct

- 2. In your professional role you shall have regard for the public health, safety and environment.
- 15. You shall not claim any level of competence that you do not possess. You shall only offer to do work or provide a service that is within your professional competence.

What are the solutions? Key Concepts underlying PSD

- Control
- Abstraction
- Packaging Solutions for Re-use
- Evidence-based development

Control

- Key issues
  - » Planning & Scheduling
  - » Risk Management
  - » Change Control
  - » Quality Assurance
  - » Cost and Quality Estimation
- Central to any large-scale project,
  - » but concrete issues and the way they're handled are often different for software projects from other types of project

Abstraction

- Modelling
  - » A software model is a representation of a software system or its context that captures selected aspects relevant to a development task
  - » *Model-based development* uses these representations to capture a software system and drive the development process
  - » UML (the Unified Modelling Language) provides a set of models useful for developing object-oriented software
- Use of Software Patterns
  - » Solutions to commonly occurring problems
  - » A way of thinking about software that abstracts from a particular implementation or context of use

Packaging Solutions for Reuse

- Object-oriented software concepts and technology
  - » Designed to make it easy to reuse designs or actual code while minimising need to make modifications
- Software Patterns (again)
- Component-based Software Technology

Evidence-based Development

- Iterative design
  - » Design process is driven by evaluation
  - » Emphases need to reflect results of evaluation in changes to requirements and design
  - » Designs are ***claims*** with respect to requirements
- User-centred design
  - » Evaluation is central
- Extreme Programming and its relatives
  - » Test driven development
    - – Design tests first
    - – Test often
  - » Rapid iterative cycle