# Lecture 2 – Extreme Programming In Practice

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

# Objectives

- To look at Extreme Programming in use
  - Justifications
  - Applicability
  - Management
- Introduction to the group project and how it will work

# Terminology

- Various methods grouped under “Agile Development”
  - Extreme Programming
  - Dynamic Systems Development Method
  - Feature Driven Development
  - Adaptive System Development
  - SCRUM
- Brought together as “Manifesto for Agile Development” www.agilemanifesto.org

# Shared, underlying values

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiations
- Responding to change over following a plan

# Why Use Agile Methods?

- Difficulty of defining requirements
- Non-linear, empirical, non-repeatable nature of software development
- Perceived failure of "conventional" methods
  - UK Economic & Social Research Council report
    - 80 – 90 % of software does not meet performance goals
    - 80% of systems are delivered late & over budget
    - 40% of developments fail or are abandoned
    - <25% of systems properly integrate business & technology
    - Only 10 – 20% meet their success criteria
  - But this tells us more about the problems with existing methods than the benefits of alternatives!

# How Choose Agile vs Plan Driven

| | Agile Features | Plan Driven Features |
|---|---|---|
| Size | Small products / teams | Large products & teams, not decomposable |
| Criticality | Unproven on safety critical projects | Methods available e.g. formal methods |
| Dynamism | Simple design & refactoring | Architecture design for stable |

# How Many Agile Projects?

- 60% of software projects involve 10 or fewer people
  - So many potential projects
- But, those small projects only account for 17% of the total value (cost) of software development
  - Not a complete solution to the “software crisis”
- Cross reference to earlier figures would be interesting
  - How many of the 60% “small” projects succeeded anyway?

# External Effects of Agile Projects

- Coordination with other project teams
  - Less well defined plans, fewer coordination points
  - Agile projects work in small "chunks"
- Human resources
  - Two programmers, one keyboard sounds unproductive!
  - Not everyone can "thrive on chaos"
- Management : two contradictory views
  - Loss of control, no plans to monitor
  - Micromanagement, frequent interaction with manager

# Project Reporting

- In plan driven projects reporting is against a plan
  - Simple to understand
  - Not necessarily a good guide to future ("80% complete")
- Agile reports
  - Small number of key dates
  - Short narrative on state of the project
  - Some metrics e.g.
    - Defects detected / cleared
    - %-age of tests passed
    - %-age of "stories" implemented
- A project continues as long as customers identify high-priority, high value work

# Group Project - Scenario

You will be provided a working program and its associated files (but not fully documented). The original author has left the organisation, although current users of the program are available and have a good understanding of the problem domain.

The organisation believes that there is commercial value in the program if it can be developed further.

You will be expected to apply Software Engineering Principles to the onward development of this program.

This is an open-ended assignment, there is no “right” answer!

# Autumn Term - Tasks

- Understand what the program does, and how it does it
  - It is approximately 750 lines of ‘C’ code
  - Examples of input and output data files are provided
  - The primary output is in the form of an HTML web page
- Find out in what ways the program could be developed
  - “expert” users available for interview
- Make a development proposal
  - Improvements / additional functionality
  - Technology
  - Development plan / approach

# Autumn Term - Deliverables

- A written report containing
  - An assessment of the program in its current state
  - A proposal for increasing the value of the program
  - A description of the tools and technologies proposed
  - A description of the proposed development methodology & techniques
  - How each individual contributed to the project
- A presentation summarising the above
  - Aimed at the organisations top management
  - The purpose is to secure “funding” for this development

# Spring Term - Tasks

- Use the tools, technologies and methodologies that you have identified to implement at least part of your proposals
- Although this will involve programming, do not forget that the purpose is to demonstrate your Software Engineering ability, not your programming ability
- E.g. Requirements Analysis, Risk Analysis, Testing, Technology Choice, Design ... ?

# Spring Term - Deliverables

- A written report containing
  - A description of what has been achieved
  - An analysis of the Software Engineering tools and techniques applied
  - An assessment of whether you achieved what you expected
  - How each individual contributed to the project

- (Hopefully) an improved version of the program
- (No presentation is required in this phase)

# Rules(!)

- This is an open-ended project, there are no right or wrong answers
- There are no restrictions on the tools, technologies or methodologies that you may use
  - Provided that they don't cost anything!
  - The means by which you make your choices are part of the project
- You may call on anyone for help of any kind
  - Provided that they don't mind!
  - That it doesn't cost anything!
  - That you acknowledge all help

# How Much Effort?

- This is up to you
- Remember that the group project is worth 30% of the coursemark
- The aim is demonstrate understanding *and analysis* of Software Engineering techniques
- In general, all members of the same project group will receive the same mark
  - Unless there are clear grounds for a greater or lesser contribution from one or more team members
- See Blackboard Calendar & Tasks for deadlines

# Next Week

- No seminar this week
- Next week’s lecture
  - Project Management Refresher
- Next week’s seminar
  - Presentation of the project material