The Crossover Project as an Introduction to Software Engineering
A. J. Cowling

Department of Computer Science
University of Sheffield

Pedagogic Framework

Curriculum Principles
- Students learn SE by doing it
- but they must be equipped for this
- this is the main importance of lectures, etc
- The curriculum structure must support these activities

Curriculum Structure
- As developed at Sheffield since 1988
- based on a major project in each year
- these form its “spine” – other courses are fitted round them
- the crossover project is the first such project

Phases in Learning SE
- Phase 0 – basic programming
- Phase 1 – software development
  - feasible solutions for functional requirements
- Phase 2 – software engineering
  - non-functional requirements and optimal solutions

Project Structure

Focus
- Introducing software development
- Working within a defined process
  - five stage waterfall
  - key aspects: activities and deliverables (ie products)
  - quality, issues and process management effectively ignored
- Working in teams, and managing time

Stages
- Requirements analysis (using UML notations and tools)
- Formal specification (using Z notation and tools)
- System design (again using UML notations)
- System implementation (using Java)
- System testing (based on the formal specification)

The Crossover Feature
- Teams swap projects at the end of each stage
- They have the same project for the first and final stages

Project Scenarios

Basic Structure
- Small business applications
  - eg theatre bookings, tennis tournament administration, menu planner, stock control
- Initially a single paragraph each

Development
- Teaching staff act as clients
- Requirements are developed in a simulated client meeting
- Clients provided with briefing material for customising requirements and playing client role

Success Factors
- Reasonable scope for customisation
- Not too complex
  - limited number of entities (about 4) and relationships (3 or 4)
  - only simple layouts of data for the user interface, and so
  - only basic widgets needed (text boxes, list boxes, buttons)

Project Management

Project Managers
- The teaching staff – also give the supporting lectures
- Each manage a set of teams
- Primarily interact with them in formal management meetings
- Also handle queries by email

Management Meetings
- Usually one per stage (two for implementation stage)
- Two purposes
  - review students’ plans for carrying out the work
- Discuss any problems identified with the work to be carried out

Stage Instructions
- Prescribe technical products to be developed
- Prescribe formal of stage report
- Require keeping of meeting minutes and timesheets
- less rigorous than for PSP

Evaluation and Conclusions

Evaluate this style of project
Conclusions for the project structure
Conclusions for operating such projects
Project Assessment

Assessment Objectives

- Establish students’ competence at technical activities;
- Establish application of personal transferable skills – teamwork & management, presentation, self-evaluation;
- Recognise individual contributions and collective work.

Assessment Basis

- Stage reports and technical deliverables: – criteria structured to match prescribed format, – components for technical sections and evaluative sections;
- Final presentation (counts as half a stage): – describes evolution of initial system from requirements, – evaluates final system against requirements;
- Individual contributions: – individual components required for technical work, – also evidence in minutes and timesheets, – may require some risk management.

Pedagogic Evaluation

Basic Constraints

- Controlled experiments are impossible;
- Hence, evaluation can only be qualitative.

Evaluation Criteria

- How well the project teaches the material, as indicated by: – the systems developed, and – the evaluations in the final presentations;
- How well it prepares the students for the following courses as indicated by the projects they carry out there.

Evidence

- Typically, one third of systems succeed, one third fail;
- Most problems occur in the design and implementation stages;
- Most students will have seen an unsuccessful project;
- Most students will have learnt from this: – the problems caused by changing requirements or scope, – the difficulty of correcting defects.

Conclusions

For the Crossover Structure

- An effective introduction to the activities of developing software;
- The structure makes the process very explicit – because of the rigid division into stages;
- The structure emphasises teamwork and management – because of the hard deadlines;
- The structure emphasises the reading of documents – as well as writing them;
- The structure lightens the emotional impact of evaluation – most of it is of other teams’ work.

For the Operation of Crossover Projects

- They should run for as long an elapsed time as possible – now 18 weeks in our structure.
- The scope of the systems must be managed carefully:
- The detail in the stage instructions must be managed carefully: – balance prescription of products and report formats, – against freedom to plan work for each stage.