SEOC2: The Big Idea

Most of you are talented individual designers of small scale systems.

Few of you have experience of large projects which involve several people over a long time span building ambitious, large systems.

SEOC2 will help you avoid being stupid when doing that.

We will cover a large number of concepts in a short time. None of these is mathematically deep and some of them, once understood, appear like “common sense” but are based on hard-gained experience.

What SEOC2 Isn't About

- Object oriented programming
  (too low level).
- A particular area of computer science
  (too parochial).
- Giving a recipe for success
  (too arrogant).
- Summarising all relevent knowledge
  (too difficult).

What SEOC2 Is About

- Learning from experience in software architectures and design patterns.
- Having a repertoire of ways of tackling large designs.
- Being able to structure and plan a large project.
- Knowing how to demonstrate to folk who matter that you have done a good job.
- Being aware of major risks to large projects and how to avoid them.
- Having some experience of tackling a difficult initial system design project as part of a team.

What We Cover

This is a guide only. I may change topics during the module if that seems appropriate.

Controlling design : Architectural and design patterns; refactoring; extreme programming; open source design.

Planning a project : estimating size; tracking changes and defects; verification, validation and testing; risk reduction.

Quality management : software standards; design and code reviews; product versus process focus.

Tricky issues : (if time allows) requirements engineering; resilient versus brittle engineering.

The Design Practical

The principal aim is to give you exposure to one of the scariest situations that can be faced by a software engineer:

- You are given a lengthy requirements document describing the operation of a complex, commercial, safety-critical business which wants a better "information system".
- You are asked to produce a competitive tender, limited in cost, to produce a first prototype system.
- If your proposal is pedestrian then you won't get the contract.
- If it is too ambitious then you may get the contract but you won't be able to deliver it.
- If it is just right then you will get the contract, deliver the prototype and be in line for developing the full system. Payday!

The Application Domain

The requirements document is **real** - produced by a software consultancy firm for a national rail company.

It is **large**, so you have to figure out how to prioritise requirements for your prototype. It is also carefully **structured**, so you must understand this structure.

The area is difficult because there is are **tensions** between the demands for availability of information; dependability of access; customisation of delivered information; and safety of rail passengers. You have to resolve these in the best way you can, and argue why your way is best.

Teamwork

The size of the task requires that you work in teams. The constraints on each team are:

- It has a leader (extra credit for this).
- It has about 5 members (negotiable as a class).
- It forms a plan for subdividing and coordinating effort in the first 2 weeks.
- It meets with me to review progress twice.

The Primary Result

Your team must produce a report:

- Length no more than 15 pages.
- Giving high level architecture.
- Giving (slightly) lower level design of critical components.
- Explaining safety issues and their resolution.
- Presenting a plan of work to produce a prototype with 60 person-months of labour over 1 year.
- Estimating the proportion contributed by each team member.

This should be delivered to me in the SEOC2 lecture on 21st February.

Assessing the Projects

Normally project proposals are reviewed by a group of experts, each of which presents recommendations to a chair who then decides on their ranking. Each of you will be an expert. I will be the chairman.

- You get copies of all the proposals.
- You assess (and rank) all except the one for your team.
- You deliver to me a 1-page summary in the SEOC2 lecture on 7th March.
- Having taken your arguments into account, I announce the winner in the last SEOC2 lecture (9th March).

Awarding of Credit

The intention is that this practical exercise (although scary and difficult) is fun, but you will also want to know what the reward will be. It counts for 30% of your module mark, divided as follows:

- 60% on the final ranking of your proposal (balanced against your proportional contribution to it].
- 30% on your 1-page assessment report.
- 10% for team leaders.

Some Hard Realities

- You can't solve the whole problem.
- You can't negotiate with me about the best strategy.
- Other teams may lie to you about their strategies.
- So you will have to do a great deal of inspired guessing.
- Clarity and insight in presentation count for as much as technical prowess.