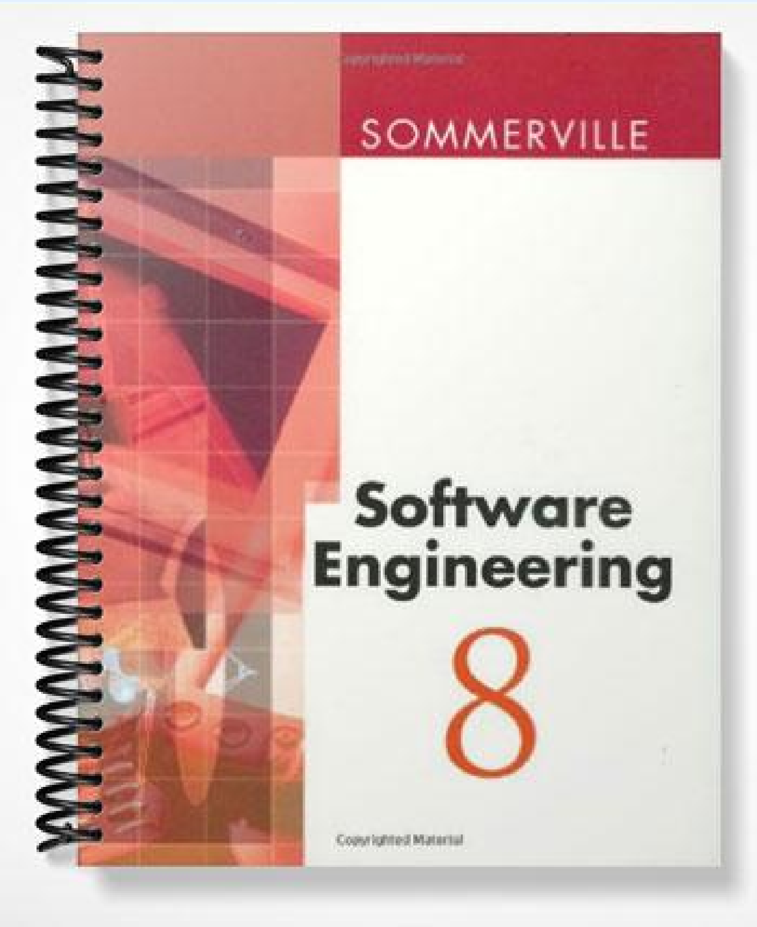
SOLUTIONS MANUAL



Software Engineering
8th edition

Solutions to selected exercises

These solutions are made available for instructional purposes only. They may only be distributed to students and it is a condition of distribution that they are only distributed by accredited instructors using 'Software Engineering, 8th edition' as a textbook.. The solutions may be made available to students on a password-protected intranet but must not be made available on a publicly-accessible WWW server.

Solutions to the exercises are organised by chapter and I have provided solutions for 6 or 7 exercises for each chapter in the book. In some cases, where the material is likely to be unfamiliar or where I have found students to have particular difficulties, a larger number of solutions are given. Overall, I have provided solutions for about 60% of the exercises. For exercises concerned with ethical issues, there are of course, no definitive solutions. For these exercises, I have included issues that might be addressed.

However, the solutions here are simply indications of what might be expected from students attempting the exercises. Many of the exercises have been deliberately designed so that they may be adapted to local situations; therefore they are not specified in a rigid way. Instructors, therefore, may use these solutions as a guide but many other possible, equally valid, solutions may also be generated.

There are still a small number of chapters where there are fewer than 6 solutions to exercises. These additional solutions will be available in the next release of this document in October 2006.

Chapter 1 Introduction

Solutions provided for Exercises 1.2, 1.3, 1.4, 1.6, 1.7 and 1.8.

1.2 The essential difference is that in generic software product development, the specification is owned by the product developer. For custom product development, the specification is owned by the customer. Of course, there may be differences in development processes but this is not necessarily the case.

1.3 For important attributes are maintainability, dependability, performance and usability. Other attributes that may be significant could be reusability (can it be reused in other applications), distributability (can it be distributed over a network of processors), portability (can it operate on multiple platforms) and inter-operability (can it work with a wide range of other software systems). Decompositions of the 4 key attributes e.g. dependability decomposes to security, safety, availability, etc. are also possible answers.

1.4 A software process is what actually goes on when software is developed. A software process model is an abstraction and simplification of a process. Process models can be used to help understand real processes and to identify which aspects of these processes could be supported by CASE tools.

1.6 Method support provided by CASE tools:

Editors for specific graphical notations used
Checking of the 'rules' and guidelines of the method
Advice to tool users on what to do next
Maintenance of a data dictionary - all names used in the system
Automatic generation of skeleton code from the system models
Generation of reports on the design

1.7 Problems and challenges for software engineering

Developing systems for multicultural use
Developing systems that can be adapted quickly to new business needs
Designing systems for outsourced development
Developing systems that are resistant to attack
Developing systems that can be adapted and configured by end-users
Finding ways of testing, validating and maintaining end-user developed systems

There are obviously lots of other problems that could be mentioned here.

1.9 Advantages of certification

- Certification is a signal to employers of some minimum level of competence.
- Certification improves the public image of the profession.
- Certification generally means establishing and checking educational standards and is therefore a mechanism for ensuring course quality.
- Certification implies responsibility in the event of disputes.

Certifying body is likely to be accepted at a national and international level as ‘speaking for the profession’.

- Certification may increase the status of software engineers and attract particularly able people into the profession.

Disadvantages of certification

- Certification tends to lead to protectionism where certified members tend not to protect others from criticism.
- Certification does not guarantee competence merely that a minimum standard was reached at the time of certification.
- Certification is expensive and will increase costs to individuals and organisations.
- Certification tends to stultify change. This is a particular problem in an area where technology developments are very rapid.

These are possible discussion points - any discussion on this will tend to be wide ranging and touch on other issues such as the nature of professionalism, etc.

Chapter 2 Computer-based system engineering

Solutions provided for Exercises 2.1, 2,2, 2.3, 2.4, 2.6, 2.7, and 2.8.

2.1 Other systems in the system's environment can have unanticipated effects because they have relationships with the system over and above whatever formal relationships (e.g. data exchange) are defined in the system specification. For example, the system may share an electrical power supply and air conditioning unit, they may be located in the same room (so if there is a fire in one system then the other will be affected) etc.

2.2 This is an inherently wicked problem because of the uncertainties associated with the problem. It is impossible to anticipate exactly when and where a disaster will occur, the numbers of people involved, the effects on the environment, the technology available to the emergency services, etc. Planning can only be in very general terms and detailed software specifications to cope with specific situations are almost impossible to write.

2.3 When a car is decommissioned, not all of its parts are worn out. Software systems can be installed in the car to monitor the different parts and to compute the lifetime which they are likely to have left. When the car is to be decommissioned, the parts which can potentially be reused can then easily be discovered.

2.4 An overall architectural description should be produced to identify sub-systems making up the system. Once these have been identified, they may be specified in parallel with other systems and the interfaces between sub-systems defined.

2.6 The key features of the solution are:

- Database with different types of data
- Video control system
- Operator console system
- River data collection
- Weather system links
- Communication control system

See Figure 2.1.

2.7 Possible issues covered in the solution might be:

- Museums are conservative places and some staff may resent the introduction of new technology.
- Existing museum staff may be asked to deal with problems of the equipment not working and may not wish to appear unable to deal with this.
- Other areas of the museum may oppose the system because they see it as diverting resources from their work.
- Different museums may have different preferred suppliers for the equipment so that all equipment used is not identical thus causing support problems.
- The new displays take up a lot of space and this displaces other displays. The maintainers of these displays may oppose the introduction of the system.
- Some museums may have no mechanism for providing technical support for the system.