

THE BCS PROFESSIONAL EXAMINATION
BCS Level 6 Professional Graduate Diploma in IT
September 2018

EXAMINERS' REPORT

Software Engineering 2

General Comments

The pass rate of less than 28% is significantly below expectation. There is evidence that the following issues (identified in previous examiner reports) continue to be significant in some centres:

1. Coverage of the syllabus. This issue is demonstrated in candidate responses to questions around topics such as Software as a Service, and Design Patterns. These subjects have limited coverage in the recommended textbooks therefore it is important that candidates have access to a broader range of resources to add breadth and depth to their knowledge of the subject at the point of delivery.
2. Examination techniques. There continues to be instances of candidates answering parts of questions without considering their indicative weightings, resulting in too detailed an answer for one component that cannot compensate for the marks lost by incomplete or "shallow" coverage of the second and subsequent components of questions. Further, candidates would benefit from reading and understanding the whole question, not just simply identify the first one or two keywords and formulating answers that do not consider the context of their use.
3. Software Engineering and Software Engineering 2. For some candidates, only basic knowledge is evidenced and a lack of advancement beyond the Waterfall model and flowcharts. Therefore, candidates should consider adopting a staged approach to their learning, by completing the BCS Level 5 Software Engineering module, before attempting BCS Level 6 Software Engineering 2. This can reduce the occurrence of using common sense interpretation of words as a substitute for depth in basic principles and concepts of software engineering.
4. Presentation. It is important that candidate responses to questions are legible, well-structured and formatted.

Question 1

A1. [Software management]

- a) It is believed that the quality of software is determined by the quality of its developers in terms of their knowledge, discipline, and commitment.

Explain the typical project manager's problem of people selection, motivation, and team effectiveness in an IT department, and discuss the extent to which the Personal Software Process model can provide a solution.

(12 marks)

- b) Explain how a software project manager would estimate and measure the software development productivity of a team.

(5 marks)

- c) Software cost estimation models are empirical models derived from data from many software projects. These have been widely used and evaluated. Discuss the relevance of software cost estimation models to an Agile software project development team.

(8 marks)

Answer Pointers

a)

The issue of **selection** focuses on experience, knowledge, and the size of the available resource pool in terms of time, internal staff members, and potential external applicants;

Motivation: The professional's personality – task-oriented; self-oriented; interaction-oriented; the instrument types: intrinsic – may touch on such things as project interest (the subject domain, the tools, the client, the team); extrinsic – may be the financial rewards, career development, travel etc.

Team effectiveness is concerned with varied and identifiable roles, communication, cohesiveness, and mutual support.

The Personal Software Process Model focuses on increasing the capability and motivation of the individual team member, within the framework of an organizational People-Capability Maturity Model. Developed and promoted by the SEI, there is emphasis on training, self-improvement, and identifiable and usable measures of performance, and rewards.

For the project manager, zero-tolerance of defects by individuals and their team brings with implicit motivation, and explicit evidence of product quality. Measures such as the number of defects per thousand lines of code may be a metric that a team/member may seek to minimize alongside an appropriate reward system. However, PSP can only be a solution if the organization is prepared to make substantial commitment and long-term investment in a capability and maturity model of this nature.

(12 marks)

b) A good answer should cover the following:

The ability to accurately estimate the productivity of a software team relies on having some measurements of the team members' past productivity in comparable projects although some estimates can be made on the basis of industry wide measurements of the average of software

developers' productivity. Productivity can be measured in lines of code produced over a fixed period of time. Other activities can also be measured, e.g. lines of documentation written, tests carried out, bugs resolved, etc. The complexity of work to be undertaken also needs to be considered, as does the level of experience and skill set of the team. Where a team has worked together on a number of projects, it may be more helpful to consider them as a single unit.

(5 marks)

c) The COCOMO model may not appear to be directly relevant to a team practicing Agile development with the Agile approach's emphasis on "people not process" in contrast to the planned based approach which emphasizes the use of well defined, standardised and managed processes. Indeed, the first version of COCOMO was based on data from industrial projects mainly in the defence and aerospace domains at a time when Agile development was not widely practised. The later COCOMO II developed in 2000 does address prototype development through its application-composition sub-model and could be applied in the case of Agile development. It estimates software size on the basis of application points which could be related to specific activities undertaken by the agile developers which result measurable outputs, e.g. lines of code, story boards, etc. Cost estimates would help define the project cost and schedule, inform investment decisions and assess any changes in technological solutions; however, the measures that need to be collected may not be possible to obtain from the agile development practitioners' in order to give credibility to their application.

(8 marks)

Examiner's Guidance Notes:

This question was attempted by nearly 97% of candidates. The pass rate was close to 38%. There is evidence that shows the following:

Part a). Many candidates sufficiently explained problems of people selection, motivation, and team effectiveness in an IT department, but only a small number discussed in the correct depth the extent to which the Personal Software Process (PSP) model can provide a solution.

Part b). Only a small number of candidates provided sufficient answers.

Part c). Many candidates provided answers that did not address the question e.g. explanation of agile methods, explanation of cost estimation techniques, etc. Only a small number discussed the relevance of established software cost estimation models (e.g. COCOMO) to an Agile software project development team.

Question 2

A2. [Advanced use of UML including OCL and use of assertions, pre- and post-conditions]

- a) The managing director of a medium financial services company recently attended a seminar on advanced UML and has asked you to:

Explain the following terms: invariant, pre- and post- conditions.

Illustrate the above terms using a customer bank account as an example.

You may assume that the following bank account operations should be specified using pre- and post- conditions:

deposit – to deposit a specified amount,

withdraw – to withdraw a specified amount if the resulting balance is greater than 0.0.

(15 marks)

- b) Briefly discuss whether the continuing development of the UML as an open standard will result in the creation of tools which:

- facilitate reverse engineering, and
- automatically generate production quality code from designs.

Justify your answer.

(10 marks)

Answer Pointers

a)

Invariant – a formal statement that must always remain true.

Pre-condition – part of an operation specification. It is a condition that must be true before the operation is executed.

Post-condition – part of an operation specification. It is a condition that must be true after the operation has executed.

(6 marks)

Formal specification using e.g. OCL is given below:

context deposit

pre: amount >0

post: balance = balance@pre + amount

--The pre-condition states that the amount to be deposited must be always greater than 0.

--The post-condition states that the new balance = the previous balance +amount

(3 marks)

context withdraw

pre: 0 < amount <= balance

post: balance = balance@pre - amount

--The pre-condition states that the amount to be withdrawn must be greater than 0 and it must be <= the current balance.

The post-condition states that the new balance = the previous balance - amount

(4 marks)

Invariant balance=>0

-The invariant states that the current balance must be always => 0

(2 marks)

b) A good answer should highlight the proliferation of tools able to generate code from designs based on UML in addition to producing UML designs from object oriented programming language source code (e.g. the Eclipse development environment). However, it is a question of whether a design in UML represent the “blueprint” for production and therefore, cannot be changed; or the means by which one gains a better understanding of the requirements and explore potential solutions before committing designs to production style toolkits.

It can be argued that the adoption of UML as a “blueprint” language is a deviation from its original purpose in design, and may result in production systems that are excessive consumers of resources, and to which insufficient time was given to the design phase.

(10 marks)

Examiner's Guidance Notes:

This question was attempted by nearly 46% of candidates. The pass rate was close to 42%. The evidence shows the following:

Part a). Overall adequate explanations of the terms were provided. However only a small number of candidates provided a 'full' specification i.e. the invariant and pre- and post-conditions for **both** operations.

Part b). Many candidates discussed reverse engineering, UML, code generation etc., but did not mention tools.

Question 3

A3. [Software metrics]

a) Give brief outlines of the following software metrics and, using specific examples, explain how each might be used to perform the quantitative assessment of software quality:

- i) architectural design metrics;
- ii) object oriented design metrics;
- iii) analysis metrics.

(15 marks)

b) Discuss the view that developments in metrics have not given any practical support to software professionals in any stage of the software development life cycle.

(10 marks)

Answer Pointers

a) A good answer should:

Give an outline of **architectural design metrics**. For example, the focus is on a framework in which a program is described by its form and structure through the collection and arrangement of modules or components, their interfaces and relationships within that context. Such metrics do not need to understand the internal implementation of modules but rather count such things as the number of interface parameters (v), and the depth or number of connections (f) to or from other modules.

An example of architectural metrics may be that for representing structural, data, and system complexity. The fan out value (f_{out}) is the total count of modules that are directly invoked by a given module i . For a module i ,

Structural complexity $S(i) = f_{out}(i)$ Data complexity $D(i) = v(i)/[f_{out}(i) + 1]$

System complexity $C(i) = S(i) + D(i)$

As each of the complexity value increases the overall architectural complexity will also increase. This increases the likelihood of problems in the integration and testing phase of development, and persistent defects in released software that may require on-going maintenance and bug-fixes.

(5 marks)

Object-oriented metrics aims to measure the effectiveness of an Object-Oriented design as well as reinforce good OO programming practice. Some of the measurable characteristics of an OO design are size, complexity, coupling, cohesion. As an example, the overall class size (S) can be determined by the total number of operations (Mi) and attributes (Aj) within the class. These include inherited and private instances.

$$S = M_i + A_j$$

A large value for S is indicative of a class with too much responsibility. This would reduce the potential for reusability of a class and would complicate its implementation and testing. Therefore, one should aspire for a lower value in class design by reducing the responsibility of a given class.

Other OO metrics can also be discussed.

(5 marks)

Analysis metrics. Project estimation metrics can be adapted and applied in the context of analysis and specification. The aim is to predict the size of the expected system and by so doing, gain insight into the effort and type of resources required.

The function point metric (FP) is derived from countable direct measures (such as input and output processes) identifiable in the requirements document and semiempirical assessment of their complexity. The unadjusted sums of FP are subject to a combination of empirically derived constants and value adjustment factors before it can be used in a meaningful way. The use of historical FP project data can greatly help in the future planning, execution, and assessment of quality.

(5 marks)

b) A good answer should recognise that measurements in certain areas of software development have made very little progress within the last few decades. This is particularly the case in the areas such as analysis, design, and implementation. However, it should also be noted that it is a relatively young science.

(2 marks)

Whilst the reliability and accuracy of such measures have had variable success, they have proved their worth to the software professional as good indicators of potential success and problem areas in software. Many of the metrics and supporting toolsets in many object-oriented paradigms have provided practical support in enhancing the design and build quality of applications as well as the rigour and scope of their testing.

(3 marks)

The problem rests mainly with the software industry itself and its marketplace, where semi-functional products with the most up-to-date technology are often preferred to fully functional and accurate products using proven technology. However, the existing behaviour may not be sustainable, and greater awareness amongst consumers will impact the industry, modifying behaviour, and resulting in a more mature marketplace and extensive professional practice. In the latter case continued research, and emphasis on practice with measurement within the education system will significantly affect future thinking and practice where software products are concerned.

(5 marks)

Examiner's Guidance Notes:

This question was attempted by nearly 59% of candidates. The pass rate was close to 35%. There is evidence of the following:

Part a). Only a small number of candidates explained correctly and were able to identify the correct architectural metrics. Object-oriented metrics were better presented, but the main problem was with analysis metrics. Many answers were irrelevant e.g. some candidates discussed fact finding techniques instead of analysis metrics; some candidates discussed general OO concepts instead of OO metrics; some candidates discussed LOC, cyclomatic complexity and software quality attributes as examples of architectural metrics.

Part b). A number of candidates provided reasonable (but brief) discussion. Some answers were irrelevant e.g. only software life cycle stages were discussed instead of metrics.

Question 4

B4. [Software reuse, Component based software engineering, Software product lines, Design patterns]

As a member of the software development team for a new production control system, you have been allocated the task of designing the software module that picks items from a production line, inspects the completeness of manufacture, and displays the state at various locations. The state information can be any one of “perfect”, “rework required”, “discard”, and “undecided”.

- a) Give a broad overview of design patterns and specify ONE pattern you might consider using for this software module. You should highlight key aspects of the choice made using diagrams and annotations based on the scenario described. **(15 marks)**
- b) Discuss the core composition of a software product line and give a brief outline of the various specialisations. **(10 marks)**

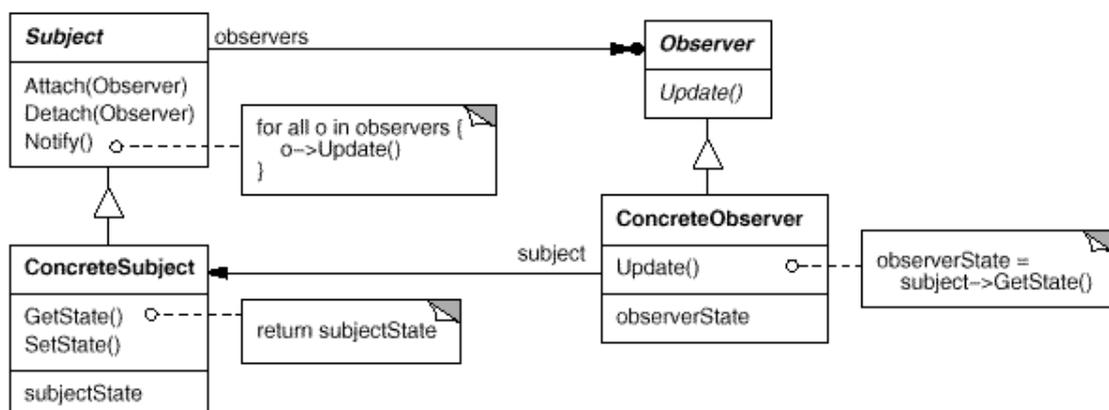
Answer Pointers

a)

A design pattern represents a tried, proven, and reusable solution to a recurrent problem within a given operational environment or context. It describes the higher-level organization of solutions to common problems.

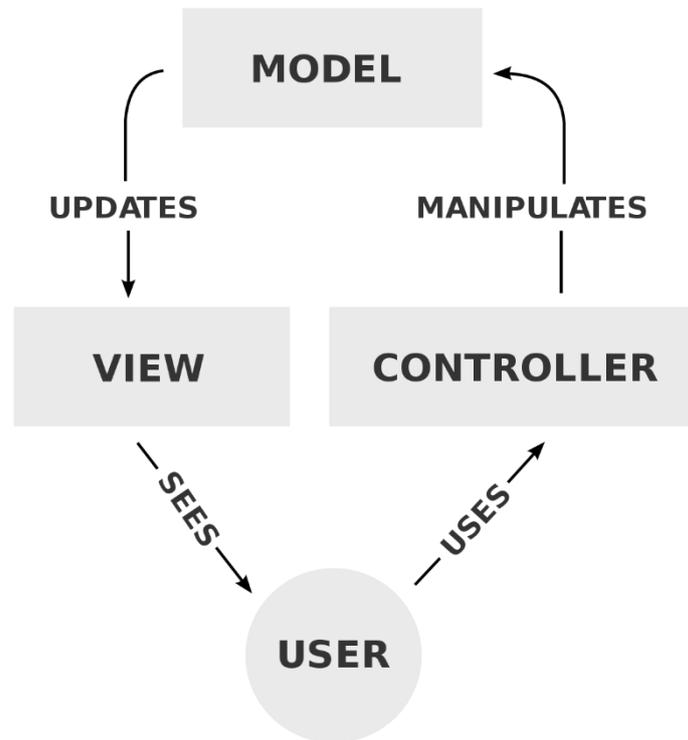
Object-oriented design patterns have been categorized as Creational, Structural, and Behavioural. The former deals with the process of object creation, the latter is concerned with interaction among classes and objects, and Structural patterns are focused on the static composition and structure of classes and objects.

The Observer pattern in the behavioural category of design patterns could be used for the construction of the software. In this scenario, the problem is to notify a changing number of objects that something has changed. The Observer pattern defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.



The software is represented as three separate classes using the Model-View-Controller (MVC) framework where the Model is the actual internal representation (or “business logic”) of the production, the View displays its status, and the Controller provides for user input and modification. In this scenario, the View would be a passive observer. Thus, it merely reports the state of production “perfect”, “rework required”, “discard”, or “undecided”. When the target object changes (state of production), the observers are notified that they should update themselves.

Decoupling the View from the Controller allow changes to the way a view responds to user input without changing its visual presentation.



(15 marks)

b)

A software product line is a set of applications with a common application-specific architecture. These include data, transaction, event, and language processing systems. In the case of transaction processing systems, core elements would include IO processing, application logic, transaction manager, and database, where the latter two may be considered elements of a Database Management System. Such systems are usually interactive and operate in real time.

(3 marks)

In the case of platform specialization transaction processing system (TPS) applications are developed for different platforms. Thus, whilst the functionality of TPS has not changed, necessary modifications have been made to the IO interface to support its use with mobile hardware technologies.

In the case of environment specialisation the TPS application is developed to operate in different environments. Thus, TPS in extreme operating environments (be it temperature or rugged landscapes) may need system components to change to reflect this.

In the case of functional and process specialization, the TPS application may be adapted for specific customers with different requirements or different business processes. This may require new functionality to be added, or existing ones to be replaced or removed.

(7 marks)

Examiner's Guidance Notes:

This question was attempted by the least number of candidates and exhibited the lowest pass rate (8%) of all the questions set.

There is evidence of the following:

Part a) there was a lack of knowledge in respect of design patterns generally. Thus, many candidates resorted to using the header information highlighting the syllabus, to construct answers on topics such as software reuse and CBSE. Furthermore, some candidates did not apply any specific design pattern and resorted to the unsuccessful application of some traditional design methods.

Part b) was not understood by candidates, with many assuming software product lines was synonymous with the Software Development Life Cycle (SDLC) and produced detailed responses on the latter.

Question 5

B5. [Software as a service, including web services and dynamic reconfiguration of Software systems]

a) Give an outline of the underlying principles of web service standards and discuss how these can provide support for the development of inter-organisational applications.

(15 marks)

b) Select a typical process model for a manufacturing business and demonstrate the design of a service-oriented system for such a company.

(10 marks)

Answer Pointers

a)

Web service is a method that facilitates the interaction of different computers over the web using universally supported protocols, such as http (HyperText Transport Protocol).

The service runs on a Web server, which is accessed by client applications. The server, having received a client request, will perform the required service and return result of this action back to the client.

The underlying principles for web service standards include the publishing and discovery of such services, their location and signatures for access, and the protocols supported. Some of the underlying standards include XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (universal description, discovery and integration).

The development of an inter-organisational application will make use of UDDI to register (as a producer) and look up (as a consumer) services with a central registry.

To allow users to interact with a service, WSDL is used to describe the published interface for that service. This will include names and aliases of services, and the number and types of parameters required, if any.

Web services are layered on top of existing, mature transfer protocols such as HTTP, SMTP, and TCP/IP. All Web Services documents are written in the XML mark-up language. Most Web services expect to be invoked using HTTP requests. Modern frameworks, such as SOAP makes use of an XML-based vocabulary for performing remote procedure calls over HTTP and other protocols. Both the request and the response are SOAP messages.

(15 marks)

b)

Three types of services may be identified for a manufacturing business such as utility services, business services, and process services. The latter could be placing orders with suppliers, goods inwards, and settling invoices.

The development of a service-oriented system for the above would require: service candidate identification, service design, and service implementation.

The services ordering, delivery, and payment can be classified as process services. The functional and non-functional service requirement should define what the service should do and any operational tolerances. For the ordering process, services may include item selection, quantity, and pricing.

The design of the service involves: the logical design of its interface in terms of defining the operations associated with the service and their parameters; the message protocol for sending and receiving requests; and the WSDL codification (representation) of service interface design.

(10 marks)

Examiner's Guidance Notes:

This was the second most popular question amongst candidates but produced one of the lowest pass rates of 13%.

There is evidence of the following:

Part a) many of the responses contained general knowledge pertaining to the web and the internet. There was little grasp of web services concept and their use in software engineering and systems development.

Part b) candidates seemed to have focussed on the concept of software process model, rather than the manufacturing process to which a service-oriented architectural approach should be applied. Many candidate responses included descriptions of models such as waterfall, prototyping, and agile approaches.