SQA Advanced Unit Specification: general information

Unit title: Software Development: Developing Small Scale Standalone Applications

Unit code: HP2N 47

Superclass: CB

Publication date: August 2017

Source: Scottish Qualifications Authority

Version: 01

Unit purpose

This Unit is designed to introduce candidates to the software development life cycle by developing and testing a small application, eg mobile, web based or PC based. The Unit should expand on the skills learned in the SQA Advanced Unit *Developing Software: Introduction* (HP1R 47) by introducing the software development life cycle, introducing user orientated design and prototyping, Introducing interactive GUI development, consolidating programming skills, expand programming skills by introducing data structures and using standard object libraries and consolidating and expanding application testing skills.

On completion of the Unit the candidate should be able to:

1 Describe a software development lifecycle model.
2 Design a small standalone interactive application.
3 Construct a small standalone interactive application.
4 Test and deploy a small standalone interactive application.

Recommended prior knowledge and skills

Access to this Unit will be at the discretion of the centre, however it would be beneficial if the candidate already possessed good written communication, critical thinking and analytical skills, either through workplace experience or training at an appropriate level.
It would also be beneficial if candidates had some prior experience of the basic programming constructs which could be evidenced by the SQA Advanced Unit *Developing Software: Introduction* (HP1R 47).

Credit points and level

2 SQA Credits at SCQF level 7: (16 SCQF credit points at SCQF level 7*)

Credit points and level

**SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from National 1 to Doctorates.*

Core Skills

Opportunities to develop aspects of Core Skills are highlighted in the Support Notes of this Unit specification.

There is no automatic certification of Core Skills or Core Skill components in this Unit.

Context for delivery

If this Unit is delivered as part of a Group Award, it is recommended that it should be taught and assessed within the subject area of the Group Award to which it contributes.

Unit specification: statement of standards

Unit title: Software Development: Developing Small Scale Standalone Applications

Unit code: HP2N 47

The sections of the Unit stating the Outcomes, Knowledge and/or Skills, and Evidence Requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the Knowledge and/or Skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Outcome 1

Describe a software development lifecycle model.

Knowledge and/or Skills

- Iterative and Incremental development
- Lifecycle Phases
- Lifecycle Activities or workflows
- Prototypes

Evidence Requirements

Closed-book assessment covering all of the Knowledge/Skills points above. Candidates must provide evidence to show that they can describe the activities undertaken in developing a small scale application.

Outcome 2

Design a small standalone interactive application.

Knowledge and/or Skills

- User interface design
- Design algorithms

Evidence Requirements

See Evidence Requirements for Outcomes 2, 3 and 4.

Outcome 3

Construct a small standalone interactive application.

Knowledge and/or Skills

- Declaring and initialising variables
- Using arithmetic, logic and Boolean operators
- Implementing control constructs
- Using array data structures appropriately
- Using event handlers
- Using modular code
- Using standard libraries
- Documenting code appropriately

Evidence Requirements

See Evidence Requirements for Outcomes 2, 3 and 4.

Outcome 4

Test and deploy a small standalone interactive application.

Knowledge and/or Skills

- Perform usability Testing
- Perform software testing
- Deploy a small application

Evidence Requirements for Outcomes 2, 3 and 4

Open-book assessment with candidates providing evidence to demonstrate their Knowledge and Skills by showing that they can develop, test and deploy a sufficiently complex small scale standalone application. The assessment should cover a representative sample of all of the knowledge points from each of the Outcomes 2, 3 and 4. The application produced can be deemed to be sufficiently complex if it:

- demonstrates an appropriate range of event handlers
- demonstrates use of a range of standard libraries
- includes an appropriate internal documentation
- demonstrates use of a least one nested control structure (eg a selection construct used within a loop construct)
- demonstrates use of a range of operators
- makes use of argument passing to user defined code ‘blocks’

The evidence for this may be provided from more than one application.

Candidates must provide a digital or paper-based record of evidence to demonstrate all Knowledge and/or Skills. Suggested approaches are outlined in the Support Notes, Guidance on the assessment of this Unit.

Unit specification: support notes

Unit title: Software Development: Developing Small Scale Standalone Applications

This part of the Unit specification is offered as guidance. The support notes are not mandatory.

While the exact time allocated to this Unit is at the discretion of the centre, the notional design length is 80 hours.

Guidance on the content and context for this Unit

This Unit has been designed as an introduction to the field of software development and as such can be used within the SQA Advanced Certificate in Computing as a taster Unit for the SQA Advanced Diploma in Computing: Software Development. It is envisaged that successful completion of the Unit should prepare candidates for undertaking the mandatory SCQF Level 8 Units in the SQA Advanced Diploma in Computing: Software Development.

The Unit can also be used as a standalone Unit for candidates with an interest in developing skills in software development. For example, it could be used to offer evening courses on developing mobile applications.

Outcome 1 introduces candidates to systems development by learning about a software development lifecycle model designed for the rapid development of small applications and comparing this to both traditional and current lifecycle models.

Iterative and Incremental development

Candidates should be taught about a traditional lifecycle model such as the waterfall model and be allowed to compare this to an iterative model such as the spiral model of systems development. They should be made aware of the problems with these models before being introduced to a current iterative and incremental model.

Lifecycle phases

The model taught should allow the candidate to learn about how the phases of a lifecycle model provide milestones which help ensure that a development can be tracked and managed. Emphasis should be placed on the requirements/planning phase.

Lifecycle activities or workflows

The activities covered will be dependent somewhat on the process introduced but candidates should be made aware of the key development skills of analysis, design, implementation and testing.

Prototypes

Candidates should be taught about both revolutionary and evolutionary prototyping and be made aware of how prototypes can be used for evaluation.

Outcomes 2, 3 and 4 give candidates the opportunity to develop and apply software development skills by designing, building and testing a series of small applications. The emphasis is on the design and construction of the application rather than requirements analysis. This reflects the real world in that many successful small applications evolve from the ideas of individuals or small groups. It also gives candidates the time to practice and consolidate the programming skills they will need to progress onto more advanced software development courses. The assessment for these Outcomes allows candidates to develop and deploy a small application either from a given brief or from their own ideas.

It is envisaged that the majority of the delivery time will be spent on developing the programming skills outlined in Outcome 3.

Outcome 2: Design a small application

Candidates should develop skills in storyboard design and be able to transpose this design into the chosen development environment. They should understand what an algorithm is and be introduced to the concept of using algorithms to design the logic of a program.

Outcome 3: Construct a small application

Declaring and initialising variables

Candidates should understand the use of variables in applications and be able to declare and initialise them appropriately. They should understand the concept of simple types and strings but, given that a number of programming languages adopt loose typing, it may or may not be appropriate to declare variables of specific types.

Using arithmetic, comparison and boolean operators

Candidates should be able to implement each of the classes of operators appropriately.

Implementing control constructs

Candidates should be able to implement both selection and loop constructs as appropriate to the programming environment used for delivery. They should also be able to implement nested control structures.

Using array data structures appropriately

Candidates should understand the concept of arrays and be able to implement them as appropriate to the programming environment used for delivery. For example if visual environments such as Scratch or AppInventor are used, lists could be used. If visual studio.net is used the common methods associated with arrays could be introduced.

Using event handlers

Candidates should be able to implement a range of event handlers and understand the concept of trigger events being used to call the handlers. Candidates should be introduced to at least five different events which must include the click event.

Using modular code

Depending on the programming environment selected candidates will be expected to be able to implement user defined code blocks, procedures or functions as appropriate. They should be able to define parameters and pass arguments to the code 'blocks' implemented.

Using standard libraries

Candidates should be able to use standard libraries to perform tasks as appropriate to the programming environment used for delivery. They should know how to pass arguments to library functions (functions, methods or procedures as appropriate). The libraries used should expand beyond the pre-defined event handlers. For example in visual environments they may use a range of pre-defined motion, sound, string or internet functions.

Documenting code appropriately

Candidates should be introduced to the concept of standardised internal documentation. The code produced should be internally documented as appropriate to the environment. For example in Scratch this may well be limited to project notes whereas in Eclipse you may well want to introduce JavaDoc.

Outcome 4: Test and deploy a small application

Perform usability testing

Candidates should be able to design, undertake and analyse a simple usability test. The test should be undertaken by at least five users who may be classroom peers.

Perform software testing

Candidates should be introduced to the concepts of test plans, test runs and testing strategies (Black and White box testing). They should be able to undertake and analyse a given test plan.

Deploy a small application

This should be a relatively straightforward task and may involve them for example deploying an app to a mobile phone, deploying an application to a website or using a deployment wizard to create a standalone installer for the application.

This Unit covers some of the skills described for a pre-entry/junior technician role in the National Occupational Standards — IT and Telecoms (2009). The main areas covered correspond to discipline 4.6 Human Computer Interaction/HCI Design, 4.7 Systems Design, 5.1 Systems Development, 5.2 Software Development and 5.3 IT/Technology Solution Testing. There are also ample opportunities within the Unit to address a range of skills at both foundation and intermediate level that are described in the National Occupational Standards for IT Users v3. The most likely areas to be covered would be IT Software Fundamentals, Using Mobile IT Devices and WS: Website Software depending on the environment chosen for delivery.

Guidance on the delivery of this Unit

This Unit has been designed to allow centres flexibility in terms of the chosen development environment used for delivery. PC applications may be developed using more traditional environments such as Visual Studio or with visual environments such as Scratch. Mobile applications may developed using high level languages such as Java and Objective C or a visual environment such as App Inventor.

Outcome 1 is more theoretical in nature and it would make sense to deliver it concurrently with the practical activities needed for the other Outcomes.

Outcome 1

This theoretical Outcome looks at traditional lifecycle models such the Waterfall model and illustrates how such models do not really support the rapid development of small applications. More appropriate iterative and incremental models such as user centred design, simpler RAD or simpler Agile models should be explored.

A possible approach would be to use a simplified version of the RAD process with three phases:

- Requirements Planning Phase
- Construction Phase
- Deployment and Testing Phase

Whatever process is selected, it is important to emphasise the incremental nature of the phases and the activities that would be iterated within the phases. Candidates should be made aware of the key development skills of analysis, design, implementation and testing.

Given that the later Outcomes do not cover the requirements planning phase, it would be sensible to explore this area in more detail, ideally with a series of short exercises that allow the candidates to develop some skills in requirements analysis.

The use of prototypes in developments could be introduced by reviewing partially constructed applications to test their usability and to try to discover additional features that users might want. This could be performed as a group activity and would also allow the candidates to learn about usability testing.

Outcomes 2, 3 and 4

These three Outcomes should be delivered using a holistic approach with the candidates working through a series of developments, gradually introducing the skills required for designing and constructing applications. It is envisaged that Outcome 3 will prove the most challenging for the majority of candidates and therefore most of the developments should be based on given designs. Testing should be introduced once candidates have successfully created a number of applications.

To enhance the learning experience, the first application developed could use a given design and make use of standard libraries including some additional features such as sound or motion triggered by a click event. This would allow the candidate to produce an application within a short period of time.

Subsequent applications could use a mixture of themes such as interactive stories and simple games and quizzes to gradually introduce the more complex design and programming concepts. Each exercise could include an additional development to challenge more competent developers.

Group work could be used for challenge exercises where the groups design and build their own applications. Each group could then devise a usability test for their application and then get other groups to run the tests. This would allow them to see how user involvement can help improve the usability of applications.

There is a wide range of new and novel development environments that can be used to quickly produce small applications.

The App Inventor (http://appinventor.mit.edu/explore/) environment could be used to teach and assess this Unit. More powerful android development environments such as Eclipse with the ADT Plugin (http://developer.android.com/index.html) could be introduced.

Guidance on the assessment of this Unit

Both of the assessments for this Unit could be undertaken using e-assessment. The closed-book assessment lends itself to a standard online objective assessment. The open-book assessment could be undertaken using an e-portfolio with links to completed applications and links to online surveys used for usability tests.

Closed-book Assessment

Evidence for all the Knowledge and/or Skills may be assessed using a representative sample of 20 multiple-choice questions. The sample should cover:

- Iterative and Incremental development
- Lifecycle Phases
- Lifecycle Activities or workflows
- Prototypes

The questions presented must change on each assessment occasion.

The assessment must be undertaken in supervised conditions. Candidates should complete this assessment within one hour. Candidates may not bring to the assessment event any notes, textbooks, handouts or other material.

Candidates must answer at least 60% of the questions correctly.

Open-book Assessment

Evidence for developing and deploying a small application could be achieved by building up a portfolio of evidence throughout the delivery of the Unit. This may include links to completed applications and links to online surveys used for usability testing. There should be some evidence that candidates have appropriately evaluated the results of the testing. At least one of the applications developed should be an application that the candidate has developed on their own either from a given scenario or from an idea that the candidate has formulated.

Online and Distance Learning

It would be feasible to develop a range of blended learning material to support distance learners. Online technology such as e-learning objects and links from virtual learning environments could be used support this type of delivery. Support for distance learners could be provided by both synchronous and asynchronous communication technologies such as the use of virtual classrooms and forums.

The closed-book assessment could be assessed using an online objective assessment and open-book assessment could be assessed using an E-portfolio. Care would need to be taken to ensure the authenticity of assessments undertaken by distance learners.

Opportunities for the use of e-assessment

E-assessment may be appropriate for some assessments in this Unit. By e-assessment we mean assessment which is supported by Information and Communication Technology (ICT), such as e-testing or the use of e-portfolios or social software. Centres which wish to use e-assessment must ensure that the national standard is applied to all candidate evidence and that conditions of assessment as specified in the Evidence Requirements are met, regardless of the mode of gathering evidence. Further advice is available in *SQA Guidelines on Online Assessment for Further Education (AA1641, March 2003).*

Opportunities for developing Core Skills

Although there is no automatic certification of Core Skills or Core Skill components in this Unit, there are opportunities for developing the Critical Thinking skills, Planning and Organising skills and Reviewing and Evaluating skills required for the Core Skill in *Problem Solving* at SCQF level 5. The open-book assessment could be used as the problem solving activity required for evidencing problem solving skills at SCQF level 6. Further information on the requirements for assessing problem solving skills at SCQF level 6 can be found at: https://www.sqa.org.uk/sqa/files_ccc/F3GD12_ProblemSolving_ASP_2009.pdf.

Equality and inclusion

This unit specification has been designed to ensure that there are no unnecessary barriers to learning or assessment. The individual needs of learners should be taken into account when planning learning experiences, selecting assessment methods or considering alternative evidence.

Further advice can be found on our website www.sqa.org.uk/assessmentarrangements.

History of changes to Unit

Version	Description of change	Date

SQA acknowledges the valuable contribution that Scotland's colleges have made to the development of SQA Advanced Qualifications.

FURTHER INFORMATION: Call SQA's Customer Contact Centre on 44 (0) 141 500 5030 or 0345 279 1000. Alternatively, complete our Centre Feedback Form.

General information for candidates

Unit title: Software Development: Developing Small Scale Standalone Applications

This Unit is designed to introduce you to the exciting world of software development. In this Unit you will be using a modern development environment to create a series of small interactive applications that could be mobile, web or PC based. As you develop these applications, you will have the opportunity to build software development skills including programming and user interface design.

You could choose to use these skills to help you develop your own applications that you could then upload into online marketplaces. You could also use the skills as a stepping stone towards a career in software development. The Unit helps prepare you for more advanced software development Units such as *Software Development: Object Orientated Programming* (HP2L 48) and *Software Development: Object Orientated Analysis and Design* (HP2M 48).

On successful completion of the Unit you should be able to:

1 Describe a software development lifecycle model.
2 Design a small standalone interactive application.
3 Construct a small standalone interactive application.
4 Test and deploy a small standalone interactive application.