Higher National Unit specification

General information

Unit title: Software Development: Analysis and Design (SCQF level 7)

Unit code: HA4C 34

Superclass:	CB
Publication date:	January 2016
Source:	Scottish Qualifications Authority
Version:	02

Unit purpose

The purpose of this Unit is to introduce learners to the analysis and design of software. It covers conventional and contemporary trends in software development methodologies including Waterfall, Prototyping, Incremental, Spiral, Rapid Application Development and Agile.

The Unit seeks to explore the types of activities that take place and the types of models that are produced during the analysis and design stages of the software development process. Learners will gain practical experience in using various analysis and design tools. They will be able to use this acquired knowledge to produce models and apply both the Waterfall and the Agile development approaches.

The Unit is suitable for learners with no previous experience or those who already have some basic knowledge of software development and want to further develop their skills. On completion of the Unit, learners will be competent in describing the analysis and design process and producing various analysis and design models. They should be ready to undertake the following Unit, or equivalent: *Software Development: Analysis and Design* (SCQF level 8).

Having completed this Unit, along with the Unit *Software Development: Implementation and Testing* (SCQF level 7), learners should be able to progress to the Unit *Software Development: Project* (SCQF level 7) or equivalent.

Higher National Unit specification: General information (cont)

Unit title: Software Development: Analysis and Design (SCQF level 7)

Outcomes

On successful completion of the Unit the learner will be able to:

1 Explain a range of analysis and design models and tools.
2 Explain conventional and contemporary approaches to the software development process.
3 Apply the Waterfall approach to software analysis and design.
4 Apply the Agile approach to software analysis and design.

Credit points and level

2 Higher National Unit credits at SCQF level 7: (16 SCQF credit points at SCQF level 7)

Recommended entry to the Unit

No previous knowledge of computer programming is required, but learners would benefit from having previous experience of using computers.

Access to this Unit is at the discretion of the centre, however it is desirable that the learner possesses good communication and problem solving skills as well as the ability to manipulate text and graphical information, gained through either workplace experience or training at an appropriate level. Knowledge and understanding of Information Systems is also desirable.

It would also be beneficial if learners had some prior experience of the basic concepts of software development, and analysis and design tools that could be evidenced by having achieved the National Unit DF2Y 11 *Software Development* (SCQF level 5) or equivalent.

Core Skills

Achievement of this Unit gives automatic certification of the following:

Complete Core Skill	Problem Solving at SCQF level 6
Core Skill component	None

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

Unit title: Software Development: Analysis and Design (SCQF level 7)

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.

The Assessment Support Pack (ASP) for this Unit provides assessment and marking guidelines that exemplify the national standard for achievement. It is a valid, reliable and practicable assessment. Centres wishing to develop their own assessments should refer to the ASP to ensure a comparable standard. A list of existing ASPs is available to download from SQA's website **(http://www.sqa.org.uk/sqa/46233.2769.html)**.

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**.

Higher National Unit specification: Statement of standards

Unit title: Software Development: Analysis and Design (SCQF level 7)

Acceptable performance in this Unit will be the satisfactory achievement of the standards set out in this part of the Unit specification. All sections of the statement of standards are mandatory and cannot be altered without reference to SQA.

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.

Learners 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

Explain a range of analysis and design models and tools.

Knowledge and/or Skills

- Describe the main characteristics of good program design
- Explain the concept of top-down design
- Describe the Agile approach to software development
- Describe the main characteristics of pseudocode
- Describe the use of structure charts
- Explain the importance of user interface design
- Explain user interface design principles
- Identify the main characteristics of Agile planning practices

Outcome 2

Explain conventional and contemporary approaches to the software development process.

Knowledge and/or Skills

- Describe the main characteristics of software development approaches including Waterfall, Prototyping, Incremental, Spiral, Rapid Application Development and Agile
- Describe the different phases of each approach
- Explain the iterative nature of the development process
- State the strengths and weaknesses of each approach
- Explain the factors affecting the choice of approach
- Describe the characteristics of Agile planning practices

Unit title: Software Development: Analysis and Design (SCQF level 7)

Outcome 3

Apply the Waterfall approach to software analysis and design.

Knowledge and/or Skills

- Describe information gathering methods
- Produce functional and non-functional requirements specifications
- Produce the user interface design
- Produce data dictionary for the whole system
- Produce the overall design using a structure chart
- Create detailed design using pseudocode

Outcome 4

Apply the Agile approach to software analysis and design.

Knowledge and/or Skills

- Select an Agile approach
- Produce the contents of the Agile product roadmap
- Capture requirements by employing user stories
- Produce product backlog
- Record the management and tracking of progress
- Present results of planning and seek feedback
- Reflect on future improvements

Evidence Requirements for this Unit

Learners will need to provide evidence to demonstrate their Knowledge and/or Skills across all Outcomes. The Evidence Requirements for this Unit will take two forms:

1 evidence of cognitive competence (for Outcomes 1, 2, and 3).
2 evidence of practical competence (for Outcomes 3 and 4).

Please note that Outcomes 1 and 2 cover only cognitive competences, Outcome 3 covers both cognitive and practical competencies. Outcome 4 covers only practical competences.

The evidence of cognitive competence will be the identifications, statements, descriptions and explanations required for Outcomes 1, 2, and part of 3. The evidence of practical competence will be the application of Outcomes 3 and 4.

Evidence is normally required for all of the Knowledge and Skills in every Outcome. This means that every Knowledge and Skills statement must be evidenced. However, sampling may be used in a specific circumstance (see below).

Higher National Unit specification: Statement of standards (cont)

Unit title: Software Development: Analysis and Design (SCQF level 7)

The amount of evidence should be the minimum consistent with the defined Knowledge and Skills. For Outcomes 3 and 4 candidates must apply each approach (Waterfall and Agile) on **at least one occasion**. The approaches may be applied to **routine problems**, which are straightforward and have no hidden complexities.

Evidence may be wholly or partly produced under controlled conditions. When evidence is produced in uncontrolled or loosely controlled conditions it must be authenticated. The Guide to Assessment provides further advice on methods of authentication.

There are **no time limitations** on the production of evidence (but see exception below). The evidence may be produced at any time during the life of the Unit. Candidates may use reference materials when undertaking assessment (but see exception below).

Sampling is permissible when the evidence of cognitive competence for Outcomes 1, 2, and 3 is produced by a test of knowledge and understanding. The test may take any form (including oral) but must be supervised, unseen and timed. The contents of the test must sample broadly and proportionately from the contents of Outcomes 1, 2 and 3 with greater weighting to Outcomes 1 and 2. Access to reference material is not appropriate for this type of assessment.

The Guidelines on Approaches to Assessment (see the Support Notes section of this specification) provides specific examples of instruments of assessment.

Higher National Unit Support Notes

Unit title: Software Development: Analysis and Design (SCQF level 7)

Unit Support Notes are offered as guidance and 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. The suggested time distribution is as follows:

Outcome 1: 16 hours
Outcome 2: 16 hours
Outcome 3: 24 hours
Outcome 4: 24 hours

Guidance on the content and context for this Unit

This Unit may be delivered as a stand-alone Unit or in combination with two other Units making up the Professional Development Award (PDA) in Software Development at SCQF level 7.

Learners should be encouraged to actively explore various resources including the Internet in order to familiarise themselves with a wide range of methodologies and tools as appropriate to each Learning Outcome. In addition, they must be aware of the internet safety, security, confidentiality and health and safety procedures of the organisation. It is also important to maintain the security and confidentiality of data and information and therefore leaners should be encouraged to back up and check for viruses on a regular basis.

The overall aim of this Unit is to introduce the learner to the software development analysis and design process. Although the focus for the Unit is on practical competencies, it also seeks to provide learners with knowledge and understanding of the conventional and contemporary trends in software development methodologies as well as the tools and the models which can be used during the analysis and design stages of the software development process.

It is anticipated that substantial amount of time will be spent on practical tasks, and although learners will probably use software that is available to them through their centre’s network. A suitable application package should be used to produce diagrams and design documents.

Unit title: Software Development: Analysis and Design (SCQF level 7)

Outcome 1

This Outcome allows the learner to gain knowledge of a range of analysis and design models and tools. Learners will be expected to identify the main characteristics of good program design, including maintainability, correctness, reusability, portability and efficiency. They should know that programs should be designed in a top-down manner, using stepwise refinement to decompose the problem into a series of subtasks.

Learners should recognise the importance of user interface design and be able to describe user interface design principles including structure, simplicity, visibility, feedback, tolerance and reuse. They should be able to describe the use of structure charts to show the breakdown of a system to its lowest manageable levels and their use in structured programming to arrange program modules into a tree. They should also be able to explain the main characteristics of pseudocode and its use in describing program code in a natural language format.

Learners should be able to describe the Agile approach to software development and identify the main characteristics of Agile planning practices, including absorbing feature requests in any order, maintaining continuous integration and minimising the time between requesting and releasing a feature.

Outcome 2

This Outcome allows the learner to gain knowledge about software development process. The learner is expected to identify various approaches used in developing software including: Waterfall, Prototyping, Incremental, Spiral, Rapid Application Development (RAD) and Agile.

Learners will become familiar with the conventional software development process, referred to as the Waterfall model, and the sequence of its well-defined stages including: analysis, design, implementation, testing, documentation, evaluation and maintenance.

Although the focus of this Unit is on the analysis and design stages, the learners are expected to recognise the other stages of the process and identify what kind of activities take place during each stage.

Learners will also have the opportunity to compare the conventional approach to other methods of developing software, including Agile. They should be made aware of the problems with the Waterfall model before being introduced to the more current iterative and incremental Agile. As each of these approaches has its own characteristics, learners will become familiar with the reasons why such approaches to the topic exist, and the advantages and disadvantages of each.

Unit title: Software Development: Analysis and Design (SCQF level 7)

Outcome 3

Learners should be able to apply the Waterfall approach to software analysis and design. They should be aware of the available information gathering methods including interviews, questionnaires, observation and examination of existing documentation.

They should be able to produce functional and non-functional requirements specifications. Functional requirements should give a description of the facility or feature required. They deal with what the software should do or provide for users and include description of the required functions, outlines of associated reports or online queries, and details of data to be held. Non-functional requirements detail constraints, targets or control mechanisms for the software. They describe how, how well or to what standard a function should be provided.

They should be able to produce a user interface which applies the principles of structure, simplicity, visibility, feedback, tolerance and reuse and produce data dictionary specifying the format and content of all data items used in the program.

They should be able to produce the overall design using a structure chart to show the breakdown of a system to its lowest manageable levels by arranging program modules into a tree and create a detailed design using pseudocode to describe program code in an English-like format.

Outcome 4

Learners should be able to apply the Agile approach to software analysis and design. They should be able to select an Agile approach, such as Scrum, Kanban, Extreme Programming (XP) or Crystal. They should be able to produce the contents of the Agile product roadmap, a high-level plan that describes how the product is likely to grow.

They should be able to capture requirements by employing user stories to define the functions the software must provide, and to facilitate requirements management. User stories capture the 'who', 'what' and 'why' of a requirement in a simple, concise way, often limited in detail by what can be hand-written on a small notecard.

They should be able to produce a product backlog, an ordered list of ‘requirements’ consisting of features, bug fixes, non-functional requirements, and whatever else needs to be done in order to deliver a working software system and record the management and tracking of progress.

They should be able to present results, seek feedback and reflect on future improvements. Some learners may find this challenging due to lack of confidence or poor presentational skills so a supportive environment should be provided.

Unit title: Software Development: Analysis and Design (SCQF level 7)

Guidance on approaches to delivery of this Unit

This Unit is a component of the PDA Software Development (SCQF) level 7. It should be delivered before, or in parallel with the Unit *Software Development: Implementation and Testing* (SCQF level 7). Both of the Units should be completed before delivery of the Unit *Software Development: Project* (level 7).

The Outcomes may be delivered in the order in which they are written. They have been written with a learning sequence in mind.

It is anticipated that the required concepts will be introduced by the teacher and reinforced by appropriate examples. The approach should be learner-centred, participative and practical.

There is significant scope in this Unit to illustrate concepts and skills with case studies of analysis and design. The majority of time in this Unit will be spent on the practical application of the theoretical aspects of the Unit.

Throughout this Unit, learner activities should relate to their vocational interests.

Guidance on approaches to assessment of this Unit

Evidence can be generated using different types of assessment. The following are suggestions only. There may be other methods that would be more suitable to learners.

The Outcomes can be assessed in a variety of ways. A conventional approach would involve the testing of knowledge through a selected response instrument (such as a multiple-choice or multiple response test). It is recommended that if this approach is adopted then all of the knowledge and understanding in this Unit is combined into a single test that samples from the knowledge domain, with an appropriate pass mark. For example, a multiple-choice or multiple response test, comprising 30 questions, each with four options (A–D). Candidates must answer at least 60% of the questions correctly. 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.

The assessment of practical skills throughout this Unit may be demonstrated in the context of a single extended task, or in the context of a number of smaller tasks. The candidate will normally demonstrate the skills during the teaching and learning activities of the Unit, rather than as separate assessment activities.

Evidence of practical competence can be stored in a portfolio. On completion of this Unit the portfolio should contain a range of evidence drawn from the Evidence Requirements for each Outcome.

If this Unit is undertaken in the workplace there may be opportunities for candidates to gather evidence in day to day workplace activities.

The practical competencies could be assessed through observation of candidate activity throughout the duration of the Unit (and recorded on an observation checklist). Practical competencies should be based on product evidence earlier.

Centres are reminded that prior verification of centre-devised assessments would help to ensure that the national standard is being met. Where learners experience a range of assessment methods, this helps them to develop different skills that should be transferable to work or further and higher education.

Opportunities for 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 learner evidence and that conditions of assessment as specified in the Evidence Requirements are met, regardless of the mode of gathering evidence. The most up-to-date guidance on the use of e-assessment to support SQA's qualifications is available at **www.sqa.org.uk/e-assessment**.

Opportunities for developing Core and other essential skills

The Unit should aid the development of Computational Thinking skills.

This Unit has the Core Skill of Problem Solving embedded in it, so when learners achieve this Unit their Core Skills profile will be updated to show that they have achieved Problem Solving at SCQF level 6.

History of changes to Unit

Version	Description of change	Date
02	Core Skill Problem Solving at SCQF level 6 embedded.	20/05/16

Additional copies of this Unit specification can be purchased from the Scottish Qualifications Authority. Please contact the Business Development and Customer Support team, telephone 0303 333 0330.

General information for learners

Unit title: Software Development: Analysis and Design (SCQF level 7)

This section will help you decide whether this is the Unit for you by explaining what the Unit is about, what you should know or be able to do before you start, what you will need to do during the Unit and opportunities for further learning and employment.

The purpose of this Unit is to introduce you to the analysis and design of computer software. It covers conventional and contemporary trends in software development methodologies including Waterfall, Prototyping, Incremental, Spiral, Rapid Application Development (RAD) and Agile.

It is desirable that you possess good communication and problem solving skills as well as the ability to manipulate text and graphical information, gained through either workplace experience or training at an appropriate level.

The Unit seeks to explore the types of activities that take place and the types of models that are produced during the analysis and design stages of the software development process. You will gain practical experience in using various analysis and design tools and be able to use this acquired knowledge to produce models and apply both the Waterfall and the Agile development approaches.

The Unit is suitable for learners with no previous experience or those who already have some basic knowledge of software development and want to further develop their skills.

You may be assessed in various ways, including multiple-choice questions relating to the theoretical knowledge covered in the Unit, and practical exercises applying the analysis and design skills learned.

On completion of the Unit, you will be competent in describing the analysis and design process and producing various analysis and design models. You should be ready to undertake the following Unit, or equivalent: *Software Development: Analysis and Design* (SCQF level 8). Along with the Unit *Software Development: Implementation and Testing* (SCQF level 7), you should be able to progress to the Unit *Software Development: Project* (SCQF level 7) or equivalent.

This Unit has the Core Skill of Problem Solving embedded in it, so when you achieve this Unit your Core Skills profile will be updated to show that you have achieved Problem Solving at SCQF level 6.