# Next Generation Higher National Unit Specification

## Professional Practice in Software Development (SCQF level 8)

**Unit code:** J7EE 48

**SCQF level:** 8 (32 SCQF credit points)

**Valid from:** session 2023 to 24

## Prototype unit specification for use in pilot delivery only (version 1.1) January 2024

This unit specification provides detailed information about the unit to ensure consistent and transparent assessment year on year.

This unit specification is for teachers and lecturers and contains all the mandatory information required to deliver and assess the unit.

This edition: January 2024 (version 1.1)

# Unit purpose

This unit provides learners with the opportunity to apply their knowledge and skills in software development to produce a software solution to a real-world problem, while demonstrating the professional behaviours and standards expected by industry.

Professional practice refers to a way of conducting software development activities to achieve certain standards, and satisfy criteria in both the process itself and its end-product. Standards and criteria include technical and non-technical aspects, such as ethics and sustainability.

The unit requires learners to participate in a project team centred around:

- collaboration, integration, and communication
- engaging in requirements acquisition, investigation, and analysis
- developing a software solution, and its design, construction, testing and deployment.

Learners document their processes and communicate their findings to an audience while conducting themselves in a professional manner. They also demonstrate the meta-skills that professional practice requires, including the ability to assess their own performance and plan for improvement.

This specialist unit is a mandatory component of Higher National Diploma (HND) Software Development. It integrates the knowledge and skills that learners on the HND programme gain in learning software development concepts and applying tools and techniques that are used by software development practitioners. This unit also contributes to whole qualification grading for HND Software Development.

Entry to the unit is at the centre’s discretion. However, we recommend that learners have completed Software Development at SCQF level 8. This includes knowledge and skills related to:

- the software development lifecycle
- object-oriented analysis and design
- object-oriented programming and data structures
- software development: testing and deployment

Progression opportunities from the unit include progressing to qualifications or degree programmes at SCQF level 9 in software development or computer science-related disciplines.

# Unit outcomes

Learners who complete this unit can:

1 explain software development methodologies
2 critically appraise project management approaches used in software development
3 use a range of software development tools
4 create a software development project plan
5 create a software design from given requirements
6 code a software application from a given design
7 create a test plan and report on test outcomes for a software application
8 deploy a software application
9 present a software solution to a client audience, detailing the entire software development process
10 evaluate their own contribution to the software development team
11 develop their meta-skills in a vocational or academic context
12 develop sustainability knowledge and understanding, and skills, in a vocational context

## Evidence requirements

Learners must provide product and performance evidence. Knowledge is inferred from the product evidence. Evidence from this unit is also used in the grading method for HND Software Development.

### Product evidence

The product evidence should take the form of a documented and working software solution.

Learners produce product evidence in the context of a sizeable software development project. The project must involve a complex system based on a real-life problem and must be team-based. Each learner must play two or more roles in the software development process, including a lead role in at least one stage. Learners must provide evidence of individual contributions in their evaluation report.

The project team must produce:

- a project plan containing a project timeline, project diary and evidence and recordings of project meetings
- a user requirements specification
- a software specification
- a design specification and any relevant diagrams
- a testing plan and results documentation
- source code, including evidence of version control
- evidence of software deployment
- an application user guide
- technical documentation
- a final report containing conclusions and recommendations

Each individual learner must produce:

- a reflection on the conduct of the project, the challenges met and overcome, and the ethical considerations of the team, including:
  - self-reflection on the conduct of the project and their role in the project delivery
  - a self-assessment of meta-skills

### Performance evidence

Learners' performance evidence is comprised of:

- a team demonstration of the final solution to the software requirement
- a team presentation of 30 minutes or more that explains the challenges met and overcome in the project, along with ethical and sustainability considerations.

The presentation must involve all team members and be made to a client audience.

You record your observations of each individual learner's participation over the course of the project, the value of their contributions, and evidence of professional behaviours. The standard of evidence should be consistent with the SCQF level of the unit.

### Outcome 11

Learners develop meta-skills in the course of doing all the units as part of this HND. In this unit, you also assess their meta-skills development as an outcome, following the evidence requirements set out below.

This meta-skills outcome is the same for the HNC (SCQF level 7) and the HND (SCQF level 8). Learners who progress from the HNC to the HND should continue to develop their meta-skills. They should gather evidence in line with the outcome requirements as they work through units and projects at HND level.

### Evidence

Learners must gather evidence that shows they have:

- self-assessed their meta-skills baseline
- created a plan for their own meta-skills development
- carried out activities to develop and demonstrate their meta-skills
- used reflective practice to monitor and assess the meta-skills they have improved and developed

Skills 4.0, a skills model to drive Scotland's future, outlines three categories of meta-skills:

- self-management
- social intelligence
- innovation

Each of these comprises four meta-skills and a number of sub-skills.

There are many interrelationships and dependencies between these skills and, at SCQF level 7 and 8, learners should focus on holistic development relevant to their vocational or academic context.

See the Educator Guide for more information.

## Outcome 12 (Learning for Sustainability)

Learners gather evidence that demonstrates they can:

- identify and describe sustainability in the context of the United Nations Sustainable Development Goals (UN SDGs)
- explain how one product or process relevant to software development could be made more sustainable and help meet the aims of at least two selected UN SDGs

# Knowledge and skills

The following table shows the knowledge and skills covered by the unit outcomes:

| Knowledge | Skills |
|---|---|
| Learners should understand:<br>♦ the roles and responsibilities in software development teams<br>♦ project management concepts and principles<br>♦ the requirements acquisition approaches<br>♦ professional standards in software design and development<br>♦ software design patterns<br>♦ software design visualisation<br>♦ validation and verification<br>♦ data structures and databases<br>♦ module and system testing<br>♦ documentation: technical and user<br>♦ version control<br>♦ technology stacks and tools<br>♦ continuous development<br>♦ continuous integration<br>♦ end-user feedback<br>♦ strategies for personal improvement<br>♦ self-assessment approaches<br>♦ meta-skills, specifically<br>— the categories of self-management, social intelligence and innovation, and associated meta-skills, as described in Skills 4.0<br>— the importance of developing meta-skills, including employability, adaptability, and effectiveness<br>— what meta-skills are most relevant to the learner’s vocational context | Learners can:<br>♦ create a project plan<br>♦ manage and monitor project progress<br>♦ meet timescales and milestones<br>♦ maintain an ongoing record of activities and decisions<br>♦ participate in team and client meetings<br>♦ derive a software specification from client requirements<br>♦ perform a feasibility study<br>♦ negotiate scope and deliverables<br>♦ apply software development methods to the design, coding, testing and deployment of a software solution<br>♦ use a range of software development tools<br>♦ use version control<br>♦ deploy software<br>♦ present a solution to a client<br>♦ demonstrate collaboration and communication skills<br>♦ evaluate their own performance in the project<br>♦ plan a strategy for meta-skills development<br>♦ implement and review plans for their meta-skills development<br>♦ assess their meta-skills development<br>♦ identify a sustainability issue in software development and suggest improvement |

| **Knowledge** | **Skills** |
|---|---|
| Learners should understand:<br>♦ approaches to developing meta-skills; in particular<br>— self-awareness: analysing preferences, strengths and weaknesses; meta-skills self-assessment<br>— goal setting and action planning<br>— reflective practice: principles of reflective practice; tools and approaches for effective reflective practice<br>♦ sustainability and UN SDGs<br>♦ sustainability as it relates to software development and potential improvements | |

# Meta-skills

Throughout this unit, learners develop meta-skills to enhance their employability in the computing sector.

## Self-management

This meta-skill includes:

- focusing: demonstrating the attention to detail that developing program code and syntax demands and that is crucial to successful coding practices
- adapting: critical reflection on the processes of the project, the role(s) that the learner has played, and the knowledge and skills gained; self-learning, encouraged by the project-based nature of the unit
- initiative: displaying independent thinking; demonstrating the self-motivation, responsibility and decision making required to reach project milestones with objectives and deliverables at each stage

## Social intelligence

This meta-skill includes:

- communicating: receiving information; empathy with other learners' thoughts, intentions and ideas
- collaborating: listening and conveying information
- leading: influencing, inspiring and motivating others; being a change catalyst

## Innovation

This meta-skill includes:

- curiosity: information sourcing; recognising problems and devising solutions
- creativity: maker mentality; imagination; visualising; contributing positively to the creative design of a software solution; application of design principles and careful analysis of requirements
- sense-making: pattern recognition; holistic thinking; analysis
- critical thinking: logical and computational thinking; decomposition; judgement

# Delivery of unit

The time required varies depending on the previous experience of individual learners. Based on 160 hours delivery and assessment time, we suggest the following distribution:

**Outcome 1** — Explain software development methodologies
(5 hours)

**Outcome 2** — Use a range of software development tools
(20 hours)

**Outcome 3** — Critically appraise project management approaches used in software development
(10 hours)

**Outcome 4** — Create a software design from given requirements
(25 hours)

**Outcome 5** — Create a software development project plan
(10 hours)

**Outcome 6** — Code a software application from a design
(50 hours)

**Outcome 7** — Create a test plan and report on test outcomes for a software application
(10 hours)

**Outcome 8** — Deploy a software application
(5 hours)

**Outcome 9** — Present a software solution to a client audience, detailing the entire software development process
(5 hours)

**Outcome 10** — Evaluate own contribution to the software development team
(5 hours)

**Outcome 11** — Develop their meta-skills in a vocational or academic context
(10 hours)

**Outcome 12** — Develop sustainability knowledge and understanding, and skills, in a vocational context
(5 hours)

# Additional guidance

The guidance in this section is not mandatory.

The unit’s general context requires learners to apply software development principles to the design and implementation of a software solution. They should do this to a significant set of requirements and demonstrate competence in working in a software development team. As the unit focuses on the project and its requirements, there is minimal formal teaching of software development concepts. However, you should cover the basics of project planning and management, along with approaches to self-assessment. Learners should build their own knowledge through their experiences in the project-based team assignment.

The core learning activity is a software development project sized for at least three learners. The team project should be a complex problem that represents a real-world situation, to actively engage learners in constructing a solution. It should be large enough to foster collaboration and communication between the team members. As learning becomes student-centred, your role is as a coach or facilitator, providing guidance, resources, and advice.

You should set milestones for the project's duration. After each milestone, the team should meet with you to discuss the corresponding deliverables and provide an opportunity for you to ask questions of individual learners and give feedback. You should meet with the learners individually to discuss their progress in the project. Assessment of deliverables focuses on the consistency between the stated software design and actual program code. You should allow changes to the design where adequate justification is provided and documents are updated.

Since the evidence created during the project represents learning, you must provide feedback that is authentic and constructive to the objectives of the problem.

Each learner must write an evaluation and reflection on their own experience with the project, analysing their actions and their team's actions. This develops their understanding of what constitutes good practice in the industry.

To accommodate the project-based learning, the learning experiences should make good use of scheduled class time and combine theoretical and technical knowledge, while allowing for extended periods of team activity and collaboration, which you must monitor and observe.

We recommend that the real-life problem that you provide for the project is changed regularly.

## Assessment for grading

You should assess the product and performance evidence produced for the unit on a pass or fail basis, according to the standards set out in the assessment requirements. You can further consider this evidence as a component of grading in the HND Software Development and apply the guidance on grading to the available evidence.

For guidance on grading, you should refer to the grading pack for HND Software Development.

# Equality and inclusion

This unit is designed to be as fair and as accessible as possible with no unnecessary barriers to learning or assessment.

You should take into account the needs of individual learners when planning learning experiences, selecting assessment methods or considering alternative evidence.

Guidance on assessment arrangements for disabled learners and/or those with additional support needs is available on the assessment arrangements web page: www.sqa.org.uk/assessmentarrangements.

# Information for learners

## Professional Practice in Software Development (SCQF level 8)

This information explains:

- what the unit is about
- what you should know or be able to do before you start
- what you need to do during the unit
- opportunities for further learning and employment

## Unit information

This unit provides you with an opportunity to apply software development principles to the design and implementation of a software solution. You also demonstrate competence in working as a team member in a software development project.

This is a specialist unit, intended for learners who are interested in contemporary practices in software development. It is particularly suitable if you have a vocational interest in computing or are considering progression to university. You should have previous experience in computer programming at SCQF level 8 or above before starting the unit. We recommend that you have previous experience in software development and the software development lifecycle.

Most of your learning comes through working with a small team to manage a given software development project. This project represents a real-world situation that allows you to become actively engaged in constructing a software solution, and it requires a team of at least three members working collaboratively to design, implement, test and deploy a software application. You understand the benefits and drawbacks to several common software development methodologies, and then manage your project through the stages of your chosen methodology for the project. You acquire competence in project management and develop communication skills suited to collaborative teamwork. You learn to handle some of the conflicts that are unavoidable in a team project, make judgements of your own and other learners' work, and take responsibility for scheduling and chairing team meetings.

You acquire knowledge of a range of tools and techniques that are used in the design, implementation and testing of software applications and in the production of documentation and visualisations for your design. Your team builds error-free, version-controlled, and secure code that follows a suitable design solution. Your team achieves this by employing a robust testing strategy and methods to get your application ready for deployment.

Furthermore, you learn the importance of creating and using internal and external software development documentation. You also learn how to use an external repository to keep software secure, and exercise version control.

Your assessment for the unit focuses on the extent to which the implemented software application aligns with the design. At each milestone in your project there is opportunity to receive feedback from the assessor. Your evidence for assessment consists of the working software as deployed, along with the required documentation and evidence of your design, test plans, test results and evaluation documents. This evidence will also contribute to the whole qualification grade for HND Software Development.

Throughout the unit, you develop meta-skills covering self-management, social intelligence and innovation along with an understanding of sustainability in this vocational context.

On completion of the unit, you may progress to more specialised topics such as software engineering, data engineering, data science or machine learning at an appropriate level. You may also progress to a computing-related vendor qualification or certification exam.

# Administrative information

**Published:** January 2024 (version 1.1)

**Superclass:** AG

## History of changes

| Version | Description of change | Date |
|---|---|---|
| 1.1 | Minor update to reflect revised grading model. | January 2024 |
| | | |
| | | |
| | | |

Note: please check SQA's website to ensure you are using the most up-to-date version of this document.