Higher National Unit Specification

General information

Unit title: Agile Development: Introduction (SCQF level 7)

Unit code: HT9H 34

Superclass:	CB
Publication date:	August 2017
Source:	Scottish Qualifications Authority
Version:	01

Unit purpose

The purpose of this unit is to introduce learners to the issues involved in designing and implementing computer systems using an Agile development framework. It is a specialist unit, intended for learners undertaking a Higher National qualification in Computing or a related area that requires an understanding of the concepts and practices underpinning Agile development. The unit allows progression to more advanced study in the design and implementation of computer systems.

The unit covers the evolution of Agile methodologies, Agile roles and core principles, and the Disciplined Agile Delivery (DAD) process framework. It also gives learners the opportunity to use Scrum framework tools. While the unit refers to computer science, it does not address the computer science aspects of the field in any depth.

The unit content relates to learners' vocational interests by examining how Agile development methodologies can be applied to systems they are familiar with, such as college administration systems and retail or banking systems.

Learners should consider the social and ethical implications of the development of computer systems, including the tension between the automation of repetitive tasks and the consequent loss of jobs. However, they should also bear in mind that the application of digital technologies has resulted in the creation of new employment opportunities.

On completion of this unit, learners should be aware of the issues involved in designing and implementing computer systems using an Agile framework.

Higher National Unit Specification: General information (cont)

Unit title: Agile Development: Introduction (SCQF level 7)

Outcomes

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

1 Describe the evolution of Agile methodology.
2 Describe Agile roles, core principles and approaches.
3 Describe the Disciplined Agile Delivery (DAD) process framework.
4 Use Scrum framework tools.

Credit points and level

1 Higher National Unit credit at SCQF level 7: (8 SCQF credit points at SCQF level 7)

Recommended entry to the unit

Learners should possess basic IT skills before commencing this unit. This may be evidenced by possession of the Core Skill in *Information and Communication Technology* at SCQF level 5 (or equivalent). Some prior exposure to project management methodologies may be useful. This could be evidenced by possession of the Higher National Unit *Project Management Methodologies: Introduction (SCQF level 7).*

Core Skills

Opportunities to develop aspects of Core Skills are highlighted in the support notes for 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.

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: Agile Development: Introduction (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

Describe the evolution of Agile methodology.

Knowledge and/or skills

♦ Software Development Life Cycle (SDLC)
♦ Waterfall methodology
♦ Plan-Do-Study-Act (PDSA)
♦ Iterative and Incremental Development (IID)
♦ Test-Driven Development (TDD)
♦ Rapid Application Development (RAD)
♦ Agile manifesto

Outcome 2

Describe Agile roles, core principles and approaches.

Knowledge and/or skills

♦ Agile principles
♦ Individual roles and responsibilities
♦ Agile approaches

Outcome 3

Describe the Disciplined Agile Delivery (DAD) process framework.

Knowledge and/or skills

♦ Hybrid approach
♦ End-to-end delivery lifecycle
♦ Modelling
♦ Documentation
♦ Governance strategies

Higher National Unit Specification: Statement of standards (cont)

Unit title: Agile Development: Introduction (SCQF level 7)

Outcome 4

Use Scrum framework tools.

Knowledge and/or skills

♦ Scrum artefacts (product backlog, sprint backlog, increment, transparency)
♦ Scrum events (sprint, daily scrum, sprint review, sprint retrospective)

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:

♦ knowledge evidence (for Outcomes 1, 2 and 3)
♦ product evidence (for Outcome 4)

The knowledge evidence will be the definitions, descriptions and explanations required for Outcomes 1, 2, and 3. Evidence is normally required for all of the associated knowledge; however, the knowledge evidence may be sampled when testing is used. When testing is used, it must be controlled in terms of location, timing and access to reference materials.

The product evidence will be artefacts demonstrating the use of Scrum tools to develop a simple solution to a given scenario as required for Outcome 4. Practical tasks should be documented in an appropriate manner, eg a report (paper or electronic), a logbook or a blog.

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. The evidence may be produced at any time during the life of the unit.

Higher National Unit support notes

Unit title: Agile Development: Introduction (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 40 hours.

Guidance on the content and context for this unit

This unit is intended for anyone with an interest in Agile development methodologies, who wishes to gain a deeper understanding of these topics. It is particularly appropriate for learners undertaking a Higher National qualification in Computing or a related area.

The aim of the unit is to provide learners with a broad knowledge of the concepts of Agile development methodologies, along with practical experience in the use of Scrum. To allow learners to perform these practical tasks, centres will require suitable resources.

Although the unit is expressed in generic terms, it should be related to a context that will be familiar to learners, eg use of Agile methodologies in the context of familiar computer systems such as retail, banking or college administration systems.

Please note that this section is not a teaching syllabus and does not seek to explain each knowledge/skills statement. This section seeks to clarify the statement of standards (within this unit specification) where it is potentially ambiguous. It also focuses on non-apparent teaching and learning issues that may be over-looked, or not emphasised, during unit delivery. As such, it is not representative of the actual time spent teaching or learning specific competences or the relative importance of each competence.

Although this unit contains a significant body of knowledge, it is recommended that it is delivered in a practical context, exemplifying the use of Agile methodologies.

During the delivery of this unit it is important that every opportunity is taken to introduce real-world examples, opportunities for whole-class and group discussion and practical demonstrations wherever possible. Concepts and terminology should be presented in context throughout the unit. Video presentations should be used where appropriate for providing an alternative explanation of a difficult topic, or as a focus for class discussion or group work. Wherever possible theoretical learning should be reinforced using practical labs/demonstrations, for example the lecturer could demonstrate the use of particular Scrum tools.

Given the theoretical nature of this unit, it is intended that a significant amount of time will be made available as a central part of the course for revision, tutorials and formative assessment exercises. Learners should be strongly encouraged to undertake further reading and opportunities for individual or group research should be provided.

Higher National Unit support notes (cont)

Unit title: Agile Development: Introduction (SCQF level 7)

Outcome 1: Describe the evolution of Agile methodology.

In the 1980s most software was developed using the Software Development Life Cycle (SDLC), a well-defined process that could be followed to deliver software to meet customer requirements. SDLC (sometimes known as the waterfall methodology) involved software designers and developers meeting with customer representatives to gather, analyse and document requirements. The process made extensive use of diagrams, such as entity-relationship, state, data flow, etc. Eventually, the customer would sign off a specification.

However, problems could occur when changes inevitably arose. By the time the system was ready for implementation, months or even years after analysis started, it was no longer what the customer needed. The business had undergone changes and the system needed to change with it.

The solution was a set of software development methodologies that emphasised quick iterative development, frequent releases of working software, regular feedback from customers and the incorporation of change into the process. These methodologies are now known Agile methodologies.

The origins of Agile go back as far as the 1930s, when Walter Shewhart proposed a methodology known as Plan-Do-Study-Act (PDSA), initially designed as a way of using short-term projects to improve quality.

In the 1940s and 1950s the US Military applied the IID methodology to the development of the X-15 jet fighter and NASA used Test Driven Development (TDD) in developing software for Project Mercury.

In the 1960s IBM recognised the advantages of using iterative development methodologies, although this had little impact at the time. During the early 1970s, IBM and TRW completed several large military projects, including the command and control software for the ballistic missile defence program using iterative methods.

In the early 90s James Martin's Rapid Application Development (RAD) aimed to reduce pre-planning and to move quickly into development, so that customers could collaborate right away with the development team by seeing a working prototype in days or weeks.

The 1990s saw the introduction of Rational Unified Process (RUP) and Scrum, followed by Extreme Programming (XP).

The Agile manifesto was written in 2001 by a group of key software developers. Their website (**http://agilemanifesto.org**) provides the following information:

Higher National Unit support notes (cont)

Unit title: Agile Development: Introduction (SCQF level 7)

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt,Ron Jeffries, Jon Kern,Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

© 2001, the above authors

This declaration may be freely copied in any form, but only in its entirety through this notice.

Agile methodologies are not well suited for all software development projects, but they have gained a strong hold in recent years and certifications are now available from a number of industry bodies including the Project Management Institute (PMI) and the Scrum alliance.

Outcome 2: Describe Agile roles, core principles and approaches.

In addition to the four Agile values noted above, Agile has 12 key principles:

♦   Customer satisfaction through early and continuous software delivery
♦   Accommodate changing requirements throughout the development process
♦   Frequent delivery of working software
♦   Collaboration between business stakeholders and developers throughout the project
♦   Support, trust and motivate the people involved
♦   Enable face-to-face interactions
♦   Working software is the primary measure of progress
♦   Agile processes to support a consistent development pace
♦   Attention to technical detail and design enhances agility
♦   Simplicity
♦   Self-organising teams encourage great architectures, requirements, and designs
♦   Regular reflections on how to become more effective

Higher National Unit support notes (cont)

Unit title: Agile Development: Introduction (SCQF level 7)

Small Agile teams normally involve at least the following roles. Names may vary according to the Agile approach selected.

♦   Team lead: usually called scrum master in Scrum and team coach or project lead in other approaches. Responsible for facilitating the team, acquiring resources and shielding the team from problems. This role often incorporates the soft skills of project management, but not the technical skills, which are distributed throughout the team.
♦   Team member: sometimes referred to as developer or programmer. Responsible for creation and delivery of software.
♦   Product owner: represents the stakeholders. Responsible for the product backlog in Scrum.
♦   Stakeholder: anyone who is a direct or indirect user, user manager, funding source, or anyone else potentially affected by the project.

Larger teams may have additional members such as technical experts or independent testers.

There is a huge range of differing approaches to Agile. Major ones include: Scrum, Extreme Programming (XP), Feature-Driven Development (FDD), Kanban, Lean, Dynamic Systems Development Method (DSDM), Crystal.

Outcome 3: Describe the Disciplined Agile Delivery (DAD) process framework.

Disciplined Agile Delivery (DAD) is a hybrid approach which extends Scrum by adding strategies from Extreme Programming (XP), Agile Modelling (AM), Unified Process (UP) and other methods. It extends the construction-based lifecycle of Agile to cover the full end-to-end delivery lifecycle, from project initiation through to product delivery.

DAD makes use of strategies derived from Agile Modelling (AM), including initial requirements and architecture envisioning, Just-in-Time (JiT) model storming, continuous documentation and others.

DAD governance generally focusses on collaborative strategies that aim to enable and motivate team members.

Outcome 4: Use Scrum framework tools.

The key artefact used in Scrum is the product backlog, an ordered list of everything that might be needed in the product. It lists all the features, functions, requirements, enhancements and fixes to be included in future releases of the product. The product backlog is never complete — it evolves as development progresses. Each item in the product backlog has a description, an order, an estimate and a value. Higher-ordered items are usually more detailed than lower-ordered items.

The sprint backlog is a subset of product backlog items selected for a given sprint. It includes a plan for delivering the product increment and accomplishing the sprint goal. The total amount of work remaining in a sprint can be summed at any time. This is normally done for each daily scrum.

Unit title: Agile Development: Introduction (SCQF level 7)

An increment is the sum of all the product backup items completed during the current sprint and the value of the increments of previous sprints. At the end of a sprint, the new increment must be 'done', meaning that it is in a useable state.

Scrum relies heavily on transparency. All major decisions are made depending on the perceived state of the artefacts. If this is incorrect, decisions may be flawed.

Scrum makes use of different types of events. The core of Scrum is the sprint, a time-boxed event of one month or less, during which a useable product increment is created. A sprint incorporates a number of sub-events including sprint planning, daily scrums, development work, sprint review and the sprint retrospective. As soon as a sprint is completed, the next one starts.

The work to be carried out during a sprint is decided by the entire sprint team, led by the sprint manager, during sprint planning. This should take no longer than eight hours for a one-month sprint and should determine what can be delivered in the resulting increment and how the work required to deliver the increment can be accomplished.

The sprint goal defines the objective to be met by a sprint through implementation of selected product backlog items which deliver a single coherent function.

The daily scrum is a 15-minute event which allows the development team to synchronise their activities and produce a plan for the next 24 hours. It should be held at the same time and place each day.

A sprint review is held at the end of the sprint to inspect the increment and amend the product backlog, leading to a revised product backlog that defines the probable product backlog items for the next sprint.

The sprint retrospective gives the development team an opportunity to inspect itself and produce a plan for improvements to be implemented during the next sprint.

Guidance on approaches to delivery of this unit

Although this unit contains a significant body of knowledge, it is recommended that it is delivered in a practical context through exemplification of the principles and practice of Agile methodologies.

It is recommended that the unit is delivered in the sequence of the outcomes, since each outcome requires the underpinning knowledge and skills of earlier outcomes. A suggested distribution of time, across the outcomes, is:

Outcome 1   8 hours
Outcome 2   8 hours
Outcome 3   12 hours
Outcome 4   12 hours.

Summative assessment should be carried out towards the end of the unit, although learners could begin to generate the evidence at an earlier stage. However, in this case, the report should not be assessed until it is complete and the learner is satisfied with it.

Higher National Unit support notes (cont)

Unit title: Agile Development: Introduction (SCQF level 7)

There are opportunities to carry out formative assessment at various stages in the life of the unit. For example, formative assessment could be carried out upon the completion of each outcome to ensure that learners have grasped the knowledge and skills contained within each outcome. This would provide assessors with an opportunity to diagnose misconceptions and intervene to remedy them before progressing to the next outcome.

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.

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.

Summative assessment should take place towards the end of unit when learners will be required to individually construct a complete unit specification on a subject of their choosing. It is recommended that this is linked to their vocational interests.

This will, most likely, comprise two assessment activities. The first instrument of assessment could be an open book written report covering all outcomes.

The report may be constructed under loosely controlled conditions. For example, parts of it may not be done under the supervision of the assessor. In this scenario, authentication would be required, which could take the form of oral questioning.

The resulting report should be assessed against defined criteria and these criteria should be known to the learner before they submit their evidence. The criteria should be based on the characteristics defined in the evidence requirements section of this unit specification.

The second instrument of assessment could be a practical task where Scrum tools should be utilised to develop a simple solution to a given scenario. The solution would be partial, no coding or testing required.

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

This unit does not offer any opportunities for developing Core Skills.

History of changes to unit

Version	Description of change	Date

General information for learners

Unit title: Agile Development: Introduction (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.

This unit aims to introduce you to the issues involved in designing and implementing computer systems using an Agile development framework. It is a specialist unit, intended for learners undertaking a Higher National qualification in Computing or a related area that requires an understanding of the concepts and practices underpinning Agile development. The unit allows progression to more advanced study in the design and implementation of computer systems.

The unit covers the evolution of Agile methodologies, Agile roles and core principles, and the Disciplined Agile Delivery (DAD) process framework. It also gives you the opportunity to use Scrum framework tools. The unit content relates to vocational interests by examining how Agile development methodologies can be applied to systems you are familiar with, such as college admin systems and retail and banking systems.

You should consider the social and ethical implications of the development of computer systems, including the tension between the automation of repetitive tasks and the consequent loss of jobs. However, you should also bear in mind that the application of digital technologies has also resulted in the creation of new employment opportunities.

On successful completion of this unit, you should be aware of the issues involved in developing computer systems using an Agile framework.

For assessment purposes you may be required to:

♦ produce an open-book written report to cover all the knowledge and understanding required.
♦ complete a practical task where Scrum tools should be utilised to develop a simple solution to a given scenario.