L3 Computing - Unit 4 - Open Systems and Community Development

Relevant LINKS

BACK TO COMPUTING UNITS **[1]**

Handbook home page **[2]**

Overview

Computer Science at Platinum Level requires the candidate to understand and participate in the open source programming community. This experience will reinforce their understanding and application of appropriate software laws and licensing principles. As a result of reviewing their work, they will be able to identify and use automated methods or alternative ways of working to improve programming and using computers. Unfamiliar aspects will require support and advice from other people.

A work activity will typically be ‘straightforward or routine’ because:

The task or context will be familiar and involve few variable aspects. The techniques used will be familiar or commonly undertaken.

Example of context – designing, planing, implementing and testing a program for controlling a physical system or solving a complex problem.

Support for the assessment of this award

Example of typical Computing work at this level (Coming Soon)

Assessor's guide to interpreting the criteria

General Information

QCF general description for Level 3 qualifications

- Achievement at Level 3 (EQF Level 4) reflects the ability to identify and use relevant understanding, methods and skills to complete tasks and address problems that, while well defined, have a measure of complexity. It includes taking responsibility for initiating and completing tasks and procedures as well as exercising autonomy and judgment within limited parameters. It also reflects awareness of different perspectives or approaches within an area of study or work.
- Use factual, procedural and theoretical understanding to complete tasks and address problems that, while well defined, may be complex and non-routine.
- Identify, select and use appropriate skills, methods and procedures.
- Use appropriate investigation to inform actions.
- Review how effective methods and actions have been.
- Take responsibility for initiating and completing tasks and procedures, including, where relevant, responsibility for supervising or guiding others.
- Exercise autonomy and judgement within limited parameters information and ideas.

Requirements

- Standards must be confirmed by a trained Level 3 Assessor or higher.
- Assessors must at a minimum record assessment judgements as entries in the online mark book on the INGOTs.org certification site.
- Routine evidence of work used for judging assessment outcomes in the candidates' records of their day to day work will be available from their e-portfolios and online work.
- Assessors should ensure that relevant web pages are available to their Account Manager on request by supply of the URL.
- When the candidate provides evidence of matching all the criteria to the specification, subject to the guidance below, the assessor can request the award using the link on the certification site. The Account Manager will request a random sample of evidence from candidates' work that verifies the assessor's judgement.
- When the Account Manager is satisfied that the evidence is sufficient to safely make an award, the candidate's success will be confirmed and the unit certificate will be printable from the web site.
- Each unit at Level 3 has recommended guided learning hours based on time required to complete by an average learner.

Assessment Method
Assessors can score each of the criteria N, L, S or H. N indicates no evidence and it is the default setting. L indicates some capability but some help still required to meet the standard. S indicates that the candidate can match the criterion to its required specification in keeping with the overall level descriptor. H indicates performance that goes beyond the expected in at least some aspects. Candidates are required to achieve at least S on all the criteria to achieve the full unit award.

Expansion of the assessment criteria

1. The candidate will understand the process of community development

1.1 I can compare and contrast the processes of software development communities

Candidates can compare the working methods of two distinct communities and explain how they are similar and how they are different.

Evidence: Written reports in portfolios.

Additional information and guidance

Communities have similarities and differences in the ways they operate. They have different release procedures for code and they operate with different licensing policies. Candidates should research two such communities and produce a final report of about 1000 words summarising their findings. The report should start with an abstract containing the headline findings and then provide a general description of each project and how it is organised, any key personalities and any key statistics - size, number of downloads a month etc. The descriptions should be used as a source for the comparisons and any judgements and conclusions. Candidates should realise that structure helps in documentation as well as code.

1.2 I can explain the principles of the Open Source Way.

Candidates be familiar with the 5 principles as defined by OpenSource.com

Evidence: from portfolios of evidence.

Additional information and guidance

The principles are:

- Community
- Rapid prototyping
- Open exchange
- Participation
- Meritocracy

CROPM might help remember them. While these principles have arisen in Open Source software development communities they also have general application in other fields. Candidates should be encouraged to consider which aspects transfer well to other fields and which do not. Software is an unusual product in that it has a development cost but virtually no cost of manufacture and distribution. Other mass produced products have most of their cost in manufacture and distribution and this gets more expensive the large the product volume. With software distribution is often referred to as viral because it can be like the spread of a virus as replication and spread has much lower barriers when compared to hardware.

1.3 I can explain the relationships between commercial and volunteer interests in a software development community.

Candidates should explain how both commercial and voluntary interests can be supported by software development communities.

Evidence: from documentation in portfolios.

Additional information and guidance

They should use research to find projects that involve commercial and volunteer interests and consider how referring to software as “commercial” or “Open Source” is misleading. Open Source can be commercialised, just not by selling licenses. Open Source code can be taken and then developed further as closed source eg as with ios, open source code can be combined with closed source elements as in the Android OS. Some licenses allow code to change from Open Source to closed eg BSD and Apache while the GPL doesn’t. Commercial companies such as Canonical work with GPL licensed software using a business model that is not dependent on selling licenses. It is a complex field with many misunderstandings. Those used to closed licenses linked to commercial interests have traditionally had a lot of trouble accepting and understanding Open Source as a business model. These business models tend to be much more about cooperation than competition but competition is not entirely absent, It is more a shift in emphasis. Many large software communities have a mixture of volunteer and commercially sponsored developers.

There are many different relationships and motivations.

1.4 I can describe Sourceforge and its role in community development.

Candidates should be familiar with Sourceforge as the main repository for open source code.

Evidence: from documentation in portfolios.

Additional information and guidance

Candidates should research Sourceforge, its projects and purpose and write a concise 500 word description of it in their portfolios. Similar lengths podcasts, videos or
other media presentations are to be encouraged.

1.5 I can explain the importance of distributed revision control systems in community software development

Candidates should be able to explain version control as important to the management of any sizeable project.

Evidence: Documentation in portfolios.

Additional information and guidance

Candidates should describe one application for distributed version control referring to projects in which it is used, why the particular application was chosen and its strengths and any weaknesses.

2. The candidate will understand licensing and intellectual property

2.1 I can explain the relationship between copyright and licensing

Candidates should demonstrate they understand that any software or documentation is automatically copyright protected on origination and the license then says how it can be used.

Evidence: internal testing, documentation in portfolios.

Additional information and guidance

Copyright is becoming more and more important as a concept as information becomes the dominant world commodity. It is essential for anyone working in computing to understand copyright. Copyright works be giving the owner of the copyright power to license their work. traditionally this was “All rights reserved” meaning no-one could do anything with the work without the copyright holders permission. Now it has been realised that it is not always in the copyright holders best interests to be as conservative as this and a much greater range of licensing has emerged. As an example, Tommy Emmanuel a virtuoso guitar player has become extremely popular as a result of videos posted on YouTube. These videos would have been removed or prevented from being uploaded if the licensing regime for copyright of the late 20th Century had been applied and Mr Emmanuel would have made a lot less money. The nature of the internet means that copyright holders realise that the model of high degrees of control and throwing a lot of money at promotion does not work so effectively in a digital world. It is complicated and each product and circumstance can change the balance of risk and reward. For this reason there is now a vast range of different licenses for different products. Software licenses can be conservative as in “All rights reserved” and determining things like having to buy a now license if you buy a new computer through to very liberal, do whatever you like with the source code but don’t sue us if anything goes wrong. Candidates should also realise that if they originate work in the context of their employment it is likely that is is the employer that owns the copyright. This is because it is the employer that is paying them to originate the work. In some cases employers will have required ALL work originated by the employee to be owned by the employer even if it was done “out of hours.” There is clearly a lot of scope for legal disputes! Some people refer to “copyright free” which would only really apply to work in the public domain. Such work has been put their by the copyright holder who has declared they have no longer any copyright interest in the work. Open Source software is NOT public domain and it is not copyright free, it just has a license that allows more freedom than is usually the case with closed source software. There are many Open Source licenses and some are more liberal than others.

2.2 I can describe and explain the freedoms associated with free and open source software.

Candidates should be able to describe free and open source software in terms of the freedoms provided to users.

Evidence: portfolios and/or internal testing.

Additional information and guidance

This is an opportunity to research the background that has led to the success of the internet and much of its supporting technologies. There is a mass of information on the internet and GNU, Richard Stallman and the Free Software Foundation are good starting points. The 4 freedoms are:

0. The freedom to run the program, for any purpose (freedom 0).

1. The freedom to study how the program works, and change i so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
2. The freedom to redistribute copies so you can help your neighbour (freedom 2).
3. The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

These freedoms are also associated with Open Source software but different licenses will stipulate different conditions. For example the GPL requires all derived software as from freedom 1 to be redistributed (freedom 3) with the same license. In other words you can't just take the code make a few changes and re-license it as closed source or non-free software. The BSD and Apache licenses allow exactly this. Apple's iOS is derived from BSD and now restricted by restrictive licensing by Apple. The Linux kernel on the other hand is licensed with the GPL. Note that free and open source software is not just about free of charge, it requires the 4 freedoms to be free and open source. In fact, you can charge for free software if you can get people to pay, the free of charge bit is a bi-product of the freedom part. Candidates should be able to support their understanding of the 4 freedoms with examples from key implementations. There are endless debates about the ethics of taking community developed code and then restricting it for individual commercial gain. On the other hand, having to develop every software application from scratch duplicating a lot of well covered ground is very inefficient and restrictive licenses also tend to generate technology monopolies. The better these issues are understood the more likely good decisions will be made in any particular set of circumstances. Certainly billions have probably been wasted globally to date because of lack of understanding of these emerging changes.

2.3 I can analyse the effects of digital technologies on the enforcement of intellectual property rights.

Candidates should be able to draw rational conclusions about the practicalities of intellectual property in a digital age.

Evidence: portfolios and/or internal tests.

Additional information and guidance

The key property of digital software technologies that makes it difficult to enforce intellectual property rights is the ease of copying it. Couple this to the fact that copies are absolutely identical with no loss of quality and “piracy” (unauthorised use of copyright materials) becomes very attractive. Publishers have tried many ways to prevent this with Digital Rights Management Technologies. The problem with DRM is that it is always possible to get round it, it adds to cost and inconvenience to users. On the other hand, without it why would publishers bother publishing anything if it was likely they would never make a return? As usual, the devil is in the detail. In some circumstances, the only way to recoup the investment is to sell licenses and in order to do that the product has to be protected even if the protection isn't perfect. In others, as with Tommy Emmanuel cited above, by making the product freely available it eliminates the need for a promotion company ie a publisher as the artist can just do it themselves. If that promotes their concerts and they sell merchandise at the concerts they still make a good living. In the software development world big changes have taken place and continue to take place as a result of the way digital technologies affect the traditional commercial and economic agents. Microsoft Windows used to be the operating system on well over 95% of computing devices in 2004 and now less than 10 years later its 20%. While this is no doubt due to the rise of smartphones and tablets, the speed of development of those is almost certainly the result of ios and Android being built on free and open source operating systems when most of the work was done and could be freely used by Apple and Google. Apple had an initial headstart and went down a closed source route with a vertical market owned entirely by them. Google went down an Open Source route with multiple suppliers. At the time of writing the growth of Android looks likely to force both Apple and Windows into niche markets, but time will tell.

There is currently massive change taking place in the economic models that are most successful in the computing world and much of this is to do with the way intellectual property rights have evolved in the context of open systems computing. There is a mass of information on this subject on the internet but analysis is often weak. Encourage candidates to be critical and analytical. Get them to debate issues and back their arguments with evidence. Larry Lessig is a very good source of insight into intellectual property and ethics in the digital age.

2.4 I can describe the advantages and disadvantages of software patents.

Candidates should demonstrate an understanding of patents in general and how these fit the software paradigm.

Evidence: Documentation in portfolios.

Additional information and guidance

A patent provides a monopoly for the patent holder for a limited time. Patents are traditionally granted for inventions and were devised before the age of software. Patents were never intended to cover books, music, artistic works or scientific laws, that is the realm of copyright. Patents have been granted for some very strange things, especially in the United States. Of course any patent can be over-turned if someone shows that the invention was not new but had already existed before the patent. This is known as prior art. The snag is that is costs money to do this and so fighting patent legal battles can be unrealistic for the small players that patents were originally designed to protect. Software is a bit of a grey area because it
is already subject to copyright and has many of the features of works that have been copyrighted. On the other hand it has some of the characteristics of invention too if it is a brand new application doing an original job. There are some real practical problems with software patents. Is it realistic to be able to patent a software routine that puts a character on a screen? Imagine if one company owned such a patent and demanded say 1 cent royalty every time any computer wrote a character to the screen. Imagine a large company saying to a new start competitor your software infringes one of our patents we are not going to tell you which or which part of your software. Stop distributing your software or we will see you in court. That could ramp up costs that bankrupt the small company. Software patents are granted in the USA but not in Europe. This then affects competition across these two markets.

Candidates should make a study of the software patent issue and as a result write a 500 word summary of the case for or against. It would be useful to use the best ones to set up a formal debate on the subject.

2.5 explain the terms trademark, copyleft, creative commons, and public domain.

Candidates should be able to explain each of the terms in the context of open systems computing.

Evidence: Documentation in portfolios.

Additional information and guidance

These terms can be learnt in the context of other work in the unit and reinforced by writing a summary of each in their portfolio documentation in their own words.

3. The candidate will understand commercial models for software development

3.1 I can describe the dual licensing model for software development

Candidates should research and provide a description of such a project.

Evidence: Documentation in portfolios.

Additional information and guidance

Dual licenses are where there are two different licenses eg one for commercial organisations who have to pay and the other that is for not for profit or personal use that is free of charge. Free of charge in itself does not mean Open Source. A dual license strategy might be used to get a big take up to generate confidence quickly and then to concentrate on the development of the non-free product once established.

An example of a large project that uses dual licensing is Oracle’s MySQL.

3.2 I can describe the software as a service model.

Candidates should research and provide a description of such a project.

Evidence: from portfolios.

Additional information and guidance

Word press is and open source example, Google Apps is a proprietary example. SaaS is is any software of specific functionality which is intended to carry through a browser over a remote feed, rather than run from a local machine or local network.

3.3 I can describe the perpetual license model for software development.

Candidates should research and provide a description of such a project.

Evidence: Portfolios of evidence.

Additional information and guidance

A perpetual license allows you to use the particular version of the software for ever. MS Windows XP has such a license. In practice the software will become unsupported so it is unlikely anyone will actually use such software for an indefinite time. When microcomputers first emerged this was the most common form of license but it has now been challenged by an increasing range of business models at least in part due to the rise of the world-wide web.

3.4 I can describe the fremium model for software development.

Candidates should research and provide a description of such a project.

Evidence: Portfolios of evidence.

Additional information and guidance

A freemium business model provides a proprietary product or service free of charge, but charges for advanced features, functionality, or virtual goods. LinkedIN is an example, The basic service is free but you can pay for additional features. Note that a lot of freemium models are also software as a service models and vice versa.

3.5 I can describe an advertising model to support software development.

Candidates should research and provide a description of such a project.

Evidence: Portfolio.

Additional information and guidance

The most obvious example is Google and its search engine. This is not so much a project as an entire company. The revenue from advertising enables Google to give away products that its competitors charge for. That gives it long term sustainable competitive advantage. Any strategically useful application that will attract people will generate advertising revenue so there is no need to charge license fees for it. Google can use its competitors to spread its advertising by simply letting them use their products.

Moderation/verification

The assessor should keep a record of assessment judgements made for each candidate and make notes of any significant issues for any candidate. They must be prepared to enter into dialogue with their Account Manager and provide their assessment records to the Account Manager through the online mark book. They should be prepared to provide evidence as a basis for their judgements through reference to candidate e-portfolios and through signed witness statements associated with the criteria matching marks in the on-line mark book. Before authorizing certification, the Account Manager must be satisfied that the assessors judgements are sound.

Source URL: https://theingots.org/community/cpl3u4oscdx

Links
[1] http://theingots.org/community/Computing_qualification_info_units
[2] https://theingots.org/community/sites/default/files/uploads/common/Handbooks/Computing/L3_OpenSystems_Computing_v2%20.pdf