Unit 27: Software Development Project

Delivery guidance

Approaching the unit

The focus of this unit is the design and development of a substantial software solution that utilises front-end and back-end processes. The unit brings together a wide-range of computing skills, building on knowledge and skills developed in other units. It is expected that the learners will already have a fundamental understanding of programming concepts before starting the unit. Learners would be expected to have completed Unit 4 Programming before beginning this unit and it is also recommended that they have completed Unit 13 Software Testing, and either Unit 6 Website Development or Unit 7 Mobile Apps Development. This unit is best scheduled later in a delivery plan for example in the second year of a two-year programme.

The emphasis of the unit is the development of highly sought-after problem-solving skills. In Learning Aim A, learners will explore project planning methodologies and consider ways in which time and team members can be utilised to ensure a project meets its aims. In Learning Aims B, C and D learners will develop effective programming skills to design and create a solution. For learning Aim A, it is recommended that learners have access to project planning software such as MS Project or Project Libre. For Other learning aims learners will need access to a suitable integrated development environment (IDE) and suitable back-end resources.

This delivery guide does not cover everything that needs to be delivered for completion of this unit but gives examples of delivery methods. You should refer to the specification for full details of all the content that needs to be covered.

Delivering the Learning aims

Learning aim A you could start with some well-known examples of successful projects and compare them with failed or poorly managed projects. Discuss the constraints on a project, such as time, cost and quality in broad terms, introducing the ideas that a project’s success can be impacted by any number of many variables. Identify some of the factors that led to either success or failure in your chosen examples, explain that in this unit learners will be developing an understanding of how computing projects, whether they are small or large scale require effective management to be a success.

Explore the common approaches to software development, identifying some of the key stages when producing computer software (e.g. Investigate, design, develop, test, review). Explore how these relate to iterative and sequential methodologies and the benefits and drawbacks of different SDLC models.

You should give learners the opportunities to explore a range of project scenarios in order ion investigate how management structures and individual staff responsibilities impact on the project as a whole, and how projects and the individuals involved can be managed. Again, it may be helpful to look at successful and less successful projects, high quality case studies and input from guest speakers/employers can be of great help to provide engaging content for learners.

After learners have had an opportunity to explore examples, they should be given practical planning tasks to prepare them for the assessment. Learners will need plenty of practice in identifying and utilising resources, scheduling and budgeting. It is recommended that they have access to industry standard project management applications (e.g. MS Project, Project Libre). If your centre does not have access to project management software, learners can use other software such as spreadsheets to create Gantt charts, and other documents, but where possible it is recommended that project management software is used, as once the learners are familiar with the interface, this can help reduce the time taken in planning activities. Programs such as Project Libre, are free and open source which removes the financial restrictions that may be posed by other software.

Learning aim B you could start with considering the concept of client needs. It would be helpful to draw on knowledge from previous units where learners will have considered functional and non-functional requirements and revisit how to break a problem down in to its component parts (decomposition) and identify ways in which these problems can be solved. Initially learners should consider ‘big-picture’ issues, and it may be helpful for learners to explore a range of different scenarios and industries to gain an appreciation of how these may impact on the needs of a client.

Provide examples of documentation from different stages of the planning and design stages so learners can identify what should be produced, and what content is required for different stakeholders. Learners should be able to investigate a problem within a range of contexts and produce design documents that solve identified problems. Where learners have programming experience, it is likely that their previous experience previously will be more heavily skewed to front-end development. You may need to spend extra time introducing back-end concepts.

Depending on which units learners have studied before this unit, and the time that has passed since, you may have to revisit design documentation such as flow charts ad pseudocode in addition to the introduction of documents learners may not have had experience of previously such as data designs and UML diagrams. You should start with smaller, less complex problems and slowly build the scale and complexity of what learners would be required to design. It may also be helpful to initially scaffold tasks by providing partially completed designs that learners can improve.

Learning aims C and **D** could benefit from being delivered in a blended fashion. Learning aim C focuses on the development of software development skills to produce complete solutions that utilise front-end and back-end processes. Learning aim D focuses on the testing and refinement of these solutions. It is important that learners appreciate that these sills are interlinked and the need to test code should not be an afterthought but a key part of the coding practice.

Provide opportunities for learners to develop practical programming skills to solve both front-end and back-end computing problems. You may need to initially start with isolated tasks to build skills, as well as possibly initially introducing the concepts of front-end and back-end separately, but you should look to integrate these as soon as possible. Start with simple problems and slowly build the scale and complexity.

Assessment model

Learning aim	Key content areas	Recommended assessment approach
A Explore project planning and development methodologies	**A1** Software development life cycles **A2** Software development project management structures and responsibilities **A3** Project management processes	A project planning activity for a substantial and realistic software development project. The planning activity should include a project plan, a cost plan and a written report containing a supported rationale for planning decisions made. An evaluation of the viability of the project.
B Design a software solution for an identified client	**B1** Understanding the problem **B2** Software project proposal **B3** Software design documentation	Analysis and design of a software solution. An analysis of context, resulting in a project proposal. Design documentation for the software solution. Record of feedback received and actions taken.
C Develop a software solution that uses front-end and back-end processes for an identified client	**C1** Creating software solutions **C2** Developing high quality software solutions	A portfolio of evidence detailing the development and testing of a software solution, which may include: • completed software solution (including initial and refined version) • copy of the software solution's source code • test documentation and user feedback • analysis of feedback and evidence of
D Test and refine a software solution	**D1** Testing software solutions **D2** Use of feedback to refine software solutions	

		refinement of solution • evaluation of the development, testing and refinement process

Assessment guidance

This unit is internally assessed. There is a maximum number of three summative assignments for this unit. Tutors should refer to the assessment guidance in the specification for specific detail, particularly in relation to the requirements for Pass, Merit and Distinction grades.

The first assignment requires the learner evaluate the viability of a proposed project. They will explore a provided scenario or case study and produce a set of planning documents. They will be required to produce a project plan, a cost plan and a rationale detailing the planning decisions they made, and provide detailed recommendations and justifications to whether the prosed project is viable or not. The authorised Assignment brief provides a sample case study that centres should use as a template and adapt. It is recommended that they alter aspects such, focus of the project, individual costings, members of staff and relative experience, available budgets etc. Centres should vary the case study for each cohort, in such a way that the viability of the project is not predicable.

Learners could produce the evidence for this in different ways in including a written formal report or a presentation to the group, which would contain carefully selected aspects of their planning documentation. If a presentation is used then assessors could use video recording combined with an observation sheet to cite which assessment criteria the learner has met, with appropriate commentary supporting the reason for awarding a particular grade. A blog or some form of audio or visual evidence would also be acceptable and would allow learners to develop their creativity, provided the information is communicated in a clear and detailed manner using appropriate language.

The second assignment requires the learner to produce deigns for complete software solution that will meet the needs of an identified client. Learners should be provided with a set of general requirements which they will use develop a detailed proposal and a full set of design documentation. The proposal and documentation should demonstrate how they will meet the identified needs and will inform the development and testing of a software solution. Finally, learners will be required to evaluate the extent to which their design will meet the needs of the client, and refine as necessary before implementing their solution. their solution meets the identified objectives.

The third assignment requires the learner to develop a software solution that implements front-end and back-end processes to meet the needs of the client. They must also provide evidence that they have effectively tested and refined their solution. Finally, learners will be required to evaluate the extent to which their solution meets the identified needs.

Getting started

This gives you a starting place for one way of delivering the unit, based around the recommended assessment approach in the specification.

Introduction
Depending on which units the learners have studied before this, you may wish to briefly look at the concepts of computational thinking and how they can use this to help solve problems, particularly how a larger project can be broken down and solved. Explain to learners that the unit has two main focuses. Learning Aim A encourages learners to look at the bigger picture and the contextual factors that can influence a computing project. It should give them an understanding that software solutions exist as part of a much larger whole. Learning Aim B onwards should give learners a solid understanding of the key knowledge, skills and behaviours required of a software developer. They should understand that even substantial solutions such as the one they will develop could be components of a much larger system. The unit should provide learners with skills and competencies to successfully enter the workplace in a junior development role or allow them to progress to the study of computer science at a higher level. Where possible incorporate practical activities to develop hands on skills that learners can apply to a range of contexts.
Learning aim A: Explore project planning and development methodologies
• Learners must understand ways in which projects are planned and the factors that influence the success of projects. They should explore a range of projects in a number of scenarios. • For A1 Spend some time discussing the overall purpose of different life cycle models. It may be useful to show examples of software designed using a structured methodology and compare these with examples of software produced without it. Use plenty of real examples and devised case studies to support learning. Initially, learners could spend a short amount of time finding examples of real projects to discuss, either by looking online or at handouts • Explore the differences between sequential and iterative lifecycles identifying use cases for each and relevant benefits and drawbacks. • For A2 and A3 learners should be able to identify resources required for a project (e.g. staff, equipment, materials) and their associated costs, such as

pro-rata costing. A good starting point is to devise a simple spreadsheet for estimating project costs given the quantity of each resource required (or the duration of each task). Once learners have understood how the budget is calculated (bottom up), they could use project management software to set up resource lists and assign these to project tasks. This is a complex and time-consuming process, so allow ample time for learners to become familiar with the software.

- Provide opportunities for learners to consider the roles and responsibilities of different stakeholders, how they may influence a project and the types of information they may require. They should also be given opportunities to explore different team structures, and roles in a range of different contexts. They should explore teams that are specifically IT focused and cross discipline teams, in order to identify characteristics and needs that may influence the success of a project or the requirements for a computing solution
- Exploration of real-life workplace documentation could be helpful here, for example, org charts detailing specific line management structures can help the learners see how different roles are connected. It can be helpful to start with the structure that the learners would be familiar with, such as the structure of the teaching and SLT of your centre.
- Learners must be able to identify and plan for all the resources required for a project (staff, equipment and materials), their associated costs, such as pro-rata costing risks, and consider any potential risks.
- As with previous tasks they could analyse examples of failed projects to establish why they failed and whether, with better planning, these issues could have been avoided. Try to give some examples where serious issues were controlled and overcome by the use of good project management strategies.
- Understanding the benefits of a successfully delivered project is equally important for the initial business proposal and the final evaluation of the project. You could supply learners with brief descriptions for a number of projects and a list of typical benefits, which they should be able to match up and justify. They should also gain some experience of calculating a return on investment, using a spreadsheet to determine whether a proposed project (given time and cost) will give a financial return over a given period.
- Provide opportunities for learners to explore budget planning, in conjunction with project aims and time scale requirements. Introduce a discussion about how timescales for task completion might be estimated. Start with examples that are relatively easy to calculate, such as how long it takes to build a wall of a given length and height (knowing the size of a brick and how many bricks can be laid per hour). Explain how similar concepts

may be used to calculate the duration of a software development project, for example if we know how long particular functions typically take to code, expected testing and integration time etc. from there you could explore factors that might impact on meeting these timescales and how we could plan for these.

- Provide learners with opportunities to complete practical planning tasks in response to given scenarios. You should first expose learners to relatively straight forward problems and slowly increase the level of difficulty.

Learning aim B: Design a software solution for an identified client

- Start by recapping skills and knowledge developed in Unit 4, reminding learners of the skills needed to decompose a problem scenario and develop correct designs.
- Give learners plenty of opportunities to decompose problems and create designs for solutions using appropriate design documentation.
- Spend time exploring the concepts of front-end and back-end design, considering how the requirements for data are related to expected outcomes for the user.
- Spend some time working on the concept of abstraction and software, wherever possible hides the complexities of the data and processes in the back end.
- After some tutor-led input into conventions for designing software and presenting proposals for solutions you could provide learners with opportunities to apply these skills to given scenarios. Initially it can be helpful to separate aspects of the design process, for example, you could provide the learners with already decomposed problems, and a defined set of requirements from which they can design the just the algorithms.
- Once they are comfortable with the individual stages then provide them with opportunities to practice the whole process
- The quality and clarity of design documents is very important to ensuring a final solution is appropriate. Provide learners with examples of good and not so good design documents that they can analyse. You could also provide learners with a completed system that they have to 'reverse engineer'. In these types of activities, the learner must use a completed system to produce what they think the design documents might have looked like. Reverse engineering tasks can help learners more clearly see the link between design and final product.
- It can also be helpful to link this learning aim with Learning Aim C. Once learners have sufficient practical skills, you could provide opportunities for

them to create systems from prewritten design documentation. Varying the quality of the documents you present them allows learners to see issues that can arise if design documentation is not of sufficient quality.
Learning aim C: Develop a software solution that uses front-end and back-end processes for an identified client
• Much of the learning for this learning aim should be in the form of practical tasks. Provide opportunities for learners to spend time developing coding skills for back-end and front-end solutions. • You could start by building on the learners' previous experience of coding, which is likely to be limited to simple front-end coding and scripting. While building the complexity of the problems they solve in the front end, introduce back-end concepts such as pulling data from relatively simple file use cases (such a .csv or JSON) and slowly build up to the use of more complex data structures and back-end programming such as SQL. • Getting the learner to build on exiting code bases can help develop understating more quickly. Providing learners with solutions that have limited functionality, to which they should add additional features, will help develop understand of larger scale systems. • As learners grow in confidence you can provide less scaffolding and get learner to produce the more and more program code themselves. Some time will need to be spent introducing use of more advanced libraries and frameworks, to add greater functionality to the learners' solutions. This can be achieved through a combination of tutor-led practical demonstrations, self-directed online learning and practical challenges • This units should where possible be linked to both LAB and LAD. You could introduce the concept of testing as an integral part of the development process and not as a separate isolated task. • Provide opportunities for learners to develop solutions for a range of scenarios and different users. • Learners should be encouraged to move beyond simplistic and basic solutions. They should be remined that this unit should prepare them for becoming a computing professional and they should strive for high quality solutions. Provide the learners with opportunities to consider how ensure given opportunities to consider how they can ensure that their software is efficient, readable, maintainable and robust. • Following tutor led input, you could provide opportunities for leaners to work collaboratively, e.g. testing and debugging each other's code, parried code reviews, paired programming. You could then get learners to identify

improvements that could be made both in terms of the solution itself and the processes the learner followed.

- Provide learners with examples of good and not so good code, exploring the factors that lead to high quality software solutions.

Learning aim D: Test and refine a software solution

- This unit should where possible be linked to LAC. You should emphasise the concept of testing as an integral part of the development process and not as a separate isolated task.
- You could introduce specific types and methods of testing by providing learners with example projects/solutions in various stages of completion and with various levels of functionality. Learners could test the solutions against a defined set of requirements and document the results. For solutions that lack functionality, the learners could use their testing to identify issues and then perform the corrective actions which can then be recorded.
- Explore how the definition of requirements, and performance indicators etc at design stage should influence the testing and review process, to ensure that the solution is being tested both for functionality and the extent to which it meets the needs of the users.
- As learners' familiarity with the testing and debugging solutions increases, provide more complicated problems and less obvious issues.
- For D1 you should explore a range of different testing methodologies and how these can be applied at different stages of the development process. You should explore the importance of test data, as well as manual and automated testing.
- Explore with learners the importance of thoroughly documenting the testing process and the importance of traceability.
- For D2 you should explore the importance of test users and how feedback can be used to refine a solution. Explore reasons why a software solution might need to be refined, and that this process is not always just about finding errors but improving the overall user experience (UX). You could use class discussion to explore the learners' experiences, get them to consider times when they have used software that is 'functional' but they did not have a quality experience. What was it about the software that resulted in poor UX. Get them to contrast this to software that do enjoy using? What are the differences?
- As with Learning Aim C, it is recommended that where possible the approach to this learning aim is practical in nature.

Details of links to other BTEC units and qualifications, and to other relevant units/qualifications

This unit links to:

- Unit 4: Programming
- Unit 13: Software Testing.

Resources

Videos

https://www.youtube.com/watch?v=cGwSm8xDSwI

Backend Development – PHP and My SQL tutorial

https://www.youtube.com/c/Freecodecamp

A collection of tutorials and range of programming languages – Front end and back end

Websites

https://www.w3schools.com/
Programming tutorials for wide range of languages

https://www.codecademy.com/
Online programming courses in a range of languages

https://cloud.google.com/docs/open-tutorials
Google cloud platform tutorials and documentation

https://aws.amazon.com/education/awseducate/
Amazon Web Services – AWS educate. Platform for learners to gain free training materials and cloud-based resources

Pearson is not responsible for the content of any external internet sites. It is essential for tutors to preview each website before using it in class so as to ensure that the URL is still accurate, relevant and appropriate. We suggest that tutors bookmark useful websites and consider enabling students to access them through the school/college intranet.