Raspberry Pi Foundation

Technical Education Networks

Supporting Technical Education Teaching:

**Curriculum Resources**

# Teaching Guide

## Topic: Project brief for Design, development and implementation

Version 1, January 2024

| | |
|---|---|
| **Route** | Digital |
| **Qualification** | T Level Technical Qualification in Digital Production, Design and Development<br>https://qualifications.pearson.com/en/qualifications/t-levels/digital-production-design-and-development.html |
| **Topic** | Project brief for Design, development and implementation |
| **Specification coverage** | PO1: analyse a problem to define requirements and acceptance criteria, aligned to user needs<br>PO2: design, implement and test software<br>PO3: change, maintain and support software<br>PO4: create solutions in a social and collaborative environment<br>PO5: discover, evaluate and apply reliable sources of knowledge<br>PO6: apply ethical principles and manage risks in line with legal and regulatory requirements when developing software |

This resource is part of a series of materials to support technical education teaching. The approach to developing the materials combines the research of Professor Kevin Orr that sets out a model for technical education alongside the Raspberry Pi Foundation's twelve pedagogy principles for teaching computing.

The curriculum development begins with the knowledge that students are working to learn and apply. Teachers draw from their subject and industry expertise, and their knowledge of their students, to make decisions about the core concepts the curriculum will focus on, how they will sequence these concepts, and the activities that are selected to support students' learning. The decisions behind the resources suggested in this topic are the result of choices made by the curriculum development team, which will be reviewed and improved by teachers' decision-making and ongoing reflection in their own circumstances.

The materials also seek to support teachers in bringing classroom and industry closer together, by providing assets that draw from authentic industry materials, and using opportunities to capture workplace practice that can be shared with students.

## HEALTH AND SAFETY

It is assumed that activities outlined in this Teaching Guide will be undertaken in suitable facilities or work areas and that good practices, appropriate use policies and procedures will be observed. Teachers should consult their employers' risk assessments before use and consider whether any modification is necessary for the particular circumstances of their own class/institution.

## Acknowledgements

We are grateful to the following individuals for their input: Naomi Johns-Dyer - Truro & Penwith College, Ashvika Malde - Collyer's, and Peter Radford - City of Stoke-On-Trent Sixth Form College.

Thank you to copyright holders for permission to use artwork, photographs and other copyrighted material: credits are included alongside their position in the materials. Every effort has been made to contact copyright holders of content reproduced in this topic.

Materials for other topics are available at: www.technicaleducationnetworks.org.uk

# Contents

# Introduction

This document for teachers outlines both the topic area covered, and approach to using the suite of resources and assets.

## Topic purpose

This brief has been designed to build students' knowledge and skills for design, development and implementation of a software solution.

The brief constitutes four tasks, which may be approached sequentially, allowing students to complete the whole brief. Alternatively, students could complete the tasks as standalone activities, to focus on one particular area or to fit the brief around industry placements. The brief could be set for students to complete independently or collaboratively.

There are also opportunities to build several essential skills that are developed during the course and general competencies for maths, English and digital.

## Industry importance

Software developers and engineers design, create, and maintain the software applications and systems that operate within digital environments. Developers would typically receive project briefs with Key Performance Indicators (KPIs) and/or user acceptance criteria for projects. Typically, development would be mapped to the project brief to ensure targets are met.

The brief covers the occupational specialist knowledge and skills required to demonstrate threshold competence for the specialism. It is important that students have the opportunity to demonstrate their understanding of:

- project methodologies;
- regulatory requirements that apply to developing software;
- how to identify and manage risk;
- common software design approaches;
- testing techniques.

> *"The practice project brief illustrates each stage of the development process and provides students the opportunity to apply their skills to a real world scenario and demonstrate key transferable skills for the digital sector."*
>
> ***James Robinson, Senior Learning Manager, Raspberry Pi Foundation***

## Industry links

Many professionals working within the digital sector are members of a professional body such as:

- BCS, The Chartered Institute for IT (www.bcs.org);
- IEEE (www.ieee.org);
- The Institution of Engineering and Technology (www.theiet.org);
- The Institution of Analysts and Programmers (www.iap.org.uk);
- Business Application Software Developers Association (www.basda.org);
- Computer & Communications Industry Association (ccianet.org).

The UK government's National Careers Service for England has a section focused on careers in the digital sector: nationalcareers.service.gov.uk/job-categories/computing-technology-and-digital

# Prior learning

It is assumed that students would have studied the theory for the core components and occupational specialism before attempting the project brief. It would also be beneficial if students have some prior understanding of project management techniques and theories. A slide deck and videos are provided alongside the brief to support students with some of these concepts, if required.

# Accessibility

The teaching materials have been designed to provide teachers with a flexible framework, including different approaches to activities, suggested consolidation activities to further embed knowledge, and adaptable study questions to assess learning. As with all resources, teachers will wish to consider the specific needs of their students when using the materials, including Special Educational Needs and Disabilities (SEND).

# Performance outcomes and specification links

| Assessment overview | Performance outcomes | Brief overview | Supporting documents | Student submission | Specification links | General Competencies |
|---|---|---|---|---|---|---|
| **Task 1:**<br>Analysing the problem and designing a solution | Students will be able to:<br>• PO1: Analyse a problem to define requirements and acceptance criteria, aligned to user needs.<br>• PO2: Design, implement and test software.<br>• PO5: Discover, evaluate and apply reliable sources of knowledge.<br>• PO6: Apply ethical principles and manage risks in line with legal and regulatory requirements when developing software. | **Proposal**<br>Students should produce a proposal which:<br>• Introduces the business context<br>• Identifies and clearly defines the problem<br>• Defines the functional and non-functional requirements of the solution<br>• Defines the key performance indicators (KPIs) of the solution<br>• Defines user acceptance criteria for the proposed solution<br>• Identifies the potential risks and how these will be mitigated<br>• Describes in detail the proposed solution | • Project brief<br>• Interface design exemplar<br>• Interface design template<br>• Algorithm design exemplar<br>• Algorithm design template<br>• Test template<br>• Project manageme nt slides | • Research to explore existing and possible solutions (in an appendix for the proposal).<br>• Proposal presented as a document that has clear headings for each area.<br>• A set of design documents that include a test strategy. | **1.1** Understand the stages of the software development life cycle and be able to apply them to digital projects.<br>**1.5** Investigate the current and potential uses of emerging technologies and how they impact on industries.<br>**2.2** Identify and manage risks that apply to software development.<br>**5** All (synoptically linked).<br>**6** All (synoptically linked). | English:<br>**E1** Convey technical information to different audiences<br>**E2** Present information and ideas<br>**E3** Create texts for different purposes and audiences<br>**E4** Summarise information/ ideas<br>**E5** Synthesise information<br>**E6** Take part in/ leading discussions |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | **Design documentation**<br>Students' design documentation should clearly explain and illustrate the proposed solution. The design documentation should include:<br>• Interface designs<br>• Algorithm designs<br>• Test strategy | | | | Maths:<br>**M2** Estimate, calculate and spot errors<br>**M4** Use rules and formulae<br>**M5** Process data<br>**M6** Understand data and risk<br>**M7** Interpret and represent with mathematical diagrams<br>**M8** Communicate using mathematics<br>**M10** Optimise work processes<br><br>Digital:<br>**D1** Use digital technology and media effectively<br>**D2** Design, create and edit documents and digital media<br>**D3** Communicate and collaborate |
| **Task 2:** Developing the solution | Students will be able to:<br>• PO2: Design, implement and test software.<br>• PO3: Change, maintain and support software.<br>• PO6: Apply ethical principles and manage risks in line with legal and regulatory requirements when developing software. | **Develop a functional prototype:**<br>Students should ensure they:<br>• Implement a solution using at least two appropriate programming languages to implement front and back end processes.<br>• Select and use appropriate packages, modules, and libraries to add functionality and compatibility to their solution<br>• Use common coding conventions and use good practice when developing their solution<br>• Follow legal and regulatory guidelines and | • Asset log exemplar<br>• Asset log template<br>• Test template exemplar<br>• Test template | • Prototype- presented as key versions for submission.<br>• Asset log – using the template provided.<br>• Test strategy – using the template provided. | **2.1** Investigate the legal and regulatory requirements that apply to developing software.<br>**2.2** Identify and manage risks that apply to software development.<br>**3** All (synoptically linked).<br>**6** All (synoptically linked). | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | standards ensuring a high-quality user experience**.**<br>**Asset log**<br>Students should gather and prepare a range of appropriate assets for their prototype. They should record the source, terms of use and a clear rationale of assets used in an asset log.<br>**Testing**<br>Students should provide evidence that they have iteratively tested and refined their solution during the development of the prototype.<br>Students should use a range of test data and conditions and demonstrate how these have been used to test the prototype.<br>Students should document the iterative development process of their prototype and update their test strategy with actual tests completed to show an understanding of how test inputs, calculations, validations and processes use appropriate test data. | | | | **D4** Process and analyse numerical data<br>**D5** Be safe and responsible online<br>**D6** Code and program |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Task 3a:** Gathering feedback to inform future development | Students will be able to:<br>• PO3: Change, maintain and support software.<br>• PO4: Create solutions in a social and collaborative environment. | **Plan for gathering feedback**<br>Students need to develop a plan on how they will gather feedback on their prototype to support the planning for next steps. They should outline the audience that feedback will be collected from, the methods that will be used, the timescales for this and any contingency planning they think may be useful This may include speaking to their peers, industry placement, employers or other stakeholders.<br>As part of this plan, they will need to develop a range of materials that will support them in gathering feedback.<br>Collect feedback or demonstrate for technical/non-technical audiences.<br>Students need to use the materials they have developed to collect feedback from both technical and non-technical audiences.<br>Students should prepare and carry out a series of demonstrations that will reassure the client that their solution meets the | • Feedback template | • Create a plan to gather feedback<br>• Design feedback form(s)<br>• Evidence of collected feedback | **3.1** Understand and evaluate the reliability of different sources of knowledge.<br>**3.2** Select and use techniques to obtain qualitative and quantitative data to be able to evaluate software solutions.<br>**4** All (synoptically linked). | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | requirements of the business need.<br>Students should record the feedback in a suitable format so that it can be shared and used to inform future development of the proposed solution. | | | | |
| **Task 3b:** Evaluating feedback to inform future development | Students will be able to:<br>• PO1: Analyse a problem to define requirements and acceptance criteria, aligned to user needs.<br>• PO2: Design, implement and test software.<br>• PO4: Create solutions in a social and collaborative environment.<br>• PO5: Discover, evaluate and apply reliable sources of knowledge. | **Evaluation**<br>Students should produce an evaluation of the effectiveness of the prototype they have developed. It should consider:<br>• The effectiveness of the assets and content used<br>• Why the chosen assets were used and why some were rejected<br>• The validity and reliability of sources used<br>• Any legal or ethical implications of assets and content used<br>• How well the solution meets the functional and non-functional requirements of the brief<br>• How well the solution meets the KPIs in the business need | • Project brief<br>• Evaluation template<br>• Project management slides | • Completed evaluation template.<br>• Presentation to demonstrate the digital solution, how it could be developed and adapted further. | **1** All (synoptically linked).<br>**2** All (synoptically linked).<br>**4** All (synoptically linked).<br>**5** All (synoptically linked). | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | • Whether or not the solution meets the user acceptance criteria<br>**Present the design**<br>Students should prepare a presentation that:<br>• Demonstrates the design solution<br>• Recommends how the solution could be developed further<br>• Explains how the design could be adapted if there are changes in:<br>  ○ The business need or processes<br>  ○ Technology and need for compatibility<br>  ○ Regulatory requirements | | | | |

# Task guidance

There are two versions of the project brief. The first provides a complete and contextualised example using the Raspberry Pi Foundation Coder Dojo project. The second is a blank template, which can be adapted by teachers to meet the needs of their students.

The project brief has been designed to support students to build their knowledge and skills for design, development and implementation of a software solution. There are four tasks which may be completed in sequential order (Task 1, followed by Task 2, and so on) or individually as stand-alone tasks to support specific areas of the occupational specialism.

## Preparation

| | |
|---|---|
| **Resources provided** | • Slide deck<br>• Practice project brief<br>• Template project brief<br>• Interface design template<br>• Interface design exemplar<br>• Algorithm design template<br>• Algorithm design exemplar<br>• Test strategy template<br>• Test strategy exemplar<br>• Test template<br>• Asset log template<br>• Asset log exemplar<br>• Digital feedback template<br>• Evaluation template |
| **Equipment needed** | Access to computers and internet access |
| **Safety factors** | None but ensure acceptable use policies are followed and equipment is maintained according to manufacturer's guidelines |
| **Prior learning** | • Students should have covered the theory for the core and occupational specialism before attempting the project brief.<br>• It would also be beneficial for students to have some prior understanding of project management techniques and theories. |
| **Common misconceptions** | • Key Performance Indicators (KPIs): students may confuse these with project requirements or deliverables.<br>• Students should be made aware that even systems that are not fully functioning or that do not meet all of the project requirements can demonstrate a good awareness of project management and user needs. For example, can students identify viable areas for development that would improve their solution? |

| | |
|---|---|
| **Accessibility** | • Seek to ensure wide representation for any visiting speakers and case studies used.<br>• The project brief has been presented in both video and text format.<br>• Exemplar versions of templates are provided, which are pre-populated to demonstrate how these documents can be used.<br>• The project management slide deck includes video and support materials to guide students through the project management stages of the project. |

## Task guide

| | |
|---|---|
| **Task 1: Research, produce a proposal and a set of design documents for a digital solution**<br>SUGGESTED TIME:<br>20 guided learning hours<br>RESOURCES:<br>• Slide deck<br>• Practice project brief<br>• Interface design template<br>• Interface design exemplar<br>• Algorithm design template<br>• Algorithm design exemplar<br>• Test strategy template<br>• Test strategy exemplar<br>• Test template | • Before introducing the client and the project brief, it is recommended that you spend time introducing students to some common project management tools and techniques. You may find it helpful to use the project management slide deck to do this. It is recommended that students have knowledge of:<br>  ○ the stages of the Systems Development Life Cycle (SDLC);<br>  ○ Key Performance Indicators (KPIs) and user acceptance criteria;<br>  ○ SMART targets;<br>  ○ planning tools such as Gantt charts;<br>  ○ risk management techniques and tools;<br>  ○ common design approaches;<br>  ○ using feedback and collaboration techniques.<br>**Proposal**<br>• Students should be given the project brief, time to read it carefully and make associated notes. You may want to get students to work in small groups or pairs to go through the project brief together. You can also use the video brief to bring the project to life for students.<br>• Students should be given time to research existing solutions that provide similar features to those needed for the project. They should present this research in a way that can be included as an appendix in their final submission. You may wish to provide students with some initial sources or examples for this research.<br>• Students should then write a proposal that demonstrates their knowledge of project management tools and meets the needs of the activities outlined in Task 1.<br>**Design documentation**<br>• Students should start by developing a set of interface designs to demonstrate what their solution will look like. The interface design exemplar can be used to show students how the template can be used and the typical level of detail required. Some students may prefer to complete these designs on paper rather than digitally.<br>• Students should then develop algorithm designs to demonstrate the main functions of the system and how the problem has been broken down into more manageable sections. The algorithm design exemplar can be used to demonstrate to students how the template can be used. Students should be familiar with algorithm design and pseudocode conventions before attempting this activity. |

| | |
|---|---|
| | • Finally, students should create a test strategy and compile a set of tests that will be used to test their solution. Students should be aware of different types of test data and should record the planned tests in the test template provided.<br>**By the end of this task, students should submit:**<br>1. research to explore existing and possible solutions (in an appendix for the proposal);<br>2. proposal (presented as a document that has clear headings for each area);<br>3. a set of design documents that includes a test strategy. |
| **Task 2: Developing a solution**<br>SUGGESTED TIME:<br>30 guided learning hours<br>RESOURCES:<br>• Slide deck<br>• Practice project brief<br>• Asset log template<br>• Asset log exemplar<br>• Test strategy template<br>• Test strategy exemplar<br>• Test template | • After students have submitted their proposals and design documentation, they can move onto developing a functional prototype. Note: if students have not completed Task 1 then it would be beneficial to provide them with design documentation and a proposal which demonstrates what they should produce for this task.<br>**Develop a functional prototype**<br>• Students should be aware of any legal and regulatory requirements that apply to them when developing their software. If they have not learnt about it previously it would be good to cover:<br>  ○ intellectual property rights and licences;<br>  ○ consumer protection;<br>  ○ data protection and privacy;<br>  ○ copyright and patents.<br>• Students should select and use **two** appropriate languages to implement the front-end and back-end solution of their prototype.<br>• Students should utilise software platforms to design, build and test their prototypes.<br>• Students should be aware of common coding conventions and practices when developing their prototypes.<br>**Asset log**<br>• Students should gather and prepare a range of appropriate assets for their prototype. They should record the source, terms of use and a clear rationale of any assets used in an asset log.<br>• The asset log exemplar can be used to demonstrate how assets can be accurately recorded throughout the development of the solution.<br>• It may also be useful to discuss how assets can be judged in terms of suitability for audience and purpose.<br>**Testing**<br>• Students should be encouraged to regularly test their prototype throughout the development stage.<br>• Students should record and evidence the outcome of these tests in the test template provided.<br>**By the end of this task, students should submit:**<br>1. prototype (presented as key versions for submission);<br>2. asset log (using the template provided);<br>3. updated test strategy (using the template provided). |

| | |
|---|---|
| **Task 3a: Gathering feedback to inform future development**<br>SUGGESTED TIME:<br>15 guided learning hours<br>RESOURCES:<br>• Slide deck<br>• Practice project brief<br>• Digital feedback template | • After students have been given sufficient time to develop their functional prototype for task 2, they need to plan and gather feedback on their solution.<br>• Note: it may be that students do not have a fully working prototype at this stage, it is important to remind them here that they should reflect on what they have managed to achieve and what potential improvements they could make to develop the prototype further.<br>**Plan for gathering feedback**<br>• Students should be aware of the uses of feedback and the different methods that can be used for collecting feedback.<br>• The project management slide deck contains a video and some guidance on how feedback can be used effectively.<br>• Students should develop a range of materials to gather feedback, you can share the feedback template with them to give them some guidance and support on this.<br>• Students should collect feedback from both technical and non-technical audiences. Students could utilise their industry placement here, if suitable.<br>**By the end of this task, students should submit:**<br>1. a plan to gather feedback on their prototype;<br>2. feedback form(s) and collect feedback on their prototype. |
| **Task 3b: Evaluating feedback to inform future development**<br>SUGGESTED TIME:<br>2 guided learning hours<br>RESOURCES:<br>• Slide deck<br>• Practice project brief<br>• Evaluation template | **Evaluation**<br>• After students have gathered feedback they should write an evaluation on the effectiveness of the prototype they have developed.<br>• Note: students should have had the opportunity to complete both task 2 and task 3a before attempting this task.<br>• An evaluation template has been provided with some sentence starters to help students in writing this evaluation and covers the main areas they should consider.<br>**Present the design**<br>• Students should be given the opportunity to present their prototype designs. Some options could include presenting their designs to:<br>  ○ the rest of the class;<br>  ○ a group of teachers or lecturers;<br>  ○ their industry placement;<br>  ○ parents or guardians;<br>  ○ an open evening or showcase-type event.<br>• Students should be encouraged to refer back to the project brief in order to demonstrate how their solution meets the needs of the business and the original user acceptance criteria.<br>• Students should be able to fully demonstrate their prototype and be able to make recommendations about how their solution could be developed further.<br>• It may be useful for the class to develop some judging criteria for the presentations and for students to provide peer feedback on the quality and content of the presentations. |

| | |
|---|---|
| | **By the end of this task students should submit:**<br>1. evaluation (using the template provided);<br>2. presentation on overall findings and to demonstrate the digital solution, including how it can be developed further and adapted. |
| **TOTAL SUGGESTED TIME:** | **67 guided learning hours** |

# Web links and resources

All weblinks and resources have been used in accordance with the owner Terms and Conditions and, where specified, permission has been granted by the owner.

Inclusion of web links or mention of external organisations within the teaching materials does not imply any affiliation with, endorsement by, or sponsorship of these external organisations.

# Terms of use and disclaimer of liability

These resources are made available subject to the Technical Education Networks programme Terms and Conditions. These can be accessed at www.technicaleducationnetworks.org.uk

The Terms and Conditions set out the legal terms and associated information relating to the teaching materials and other assets produced as part of the Technical Education Networks programme. The Terms and Conditions may be updated from time to time, please ensure you have read and understood them each time you access the resource.