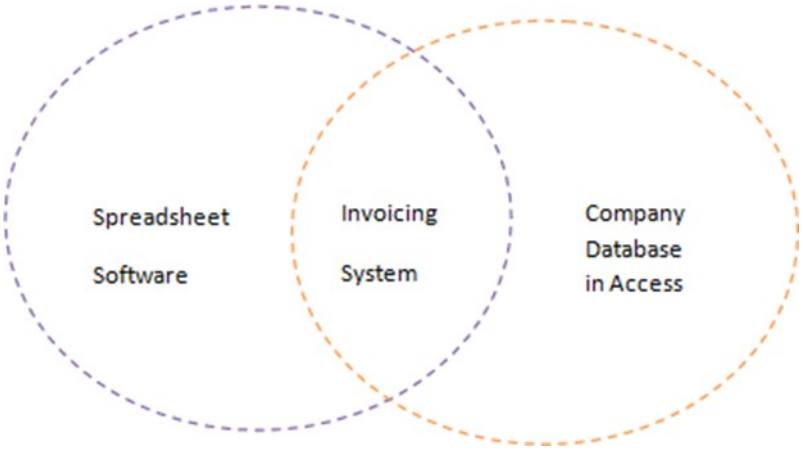
Unit 14: Customising and Integrating Applications

Delivery guidance

In business situations there will be times when a problem does not require a bespoke programmed solution because a solution could be achieved through creating customised or integrated systems using existing packages such as Microsoft Access or Excel and using VBA to provide a customised user interface that uses the functionality of the packages in a new way.

The main advantage of using existing software and its associated data files is that there is no need to migrate data from one system to another which can be expensive and time- consuming to achieve – the functionality of the software can be used as normal. For example,



Databases have limited mathematical and analytical capabilities and spreadsheets have limited data storage functionality. Creating a 'front end' using a programming language could produce an interface that allows the user to produce customer invoices using data held in the company database, via the spreadsheet software.

This makes customising a less expensive alternative to designing and programming whole new solutions.

Customisation at its simplest level could include using functionality such as paste link to create spreadsheets that are updated on open. At its most complex form, it could be a Customer Relationship Management (CRM) system that manages the data belonging to a group of companies owned by the same parent organisation.

In this unit, learners will explore customisation and integration using a range of tools and techniques. As there is a programming element, this unit would benefit from being taught alongside or following Unit 4: *Programming*.

Approaching the unit

This unit should be taught practically and this is an opportunity to involve employers, particularly small businesses, who could provide scenarios and case studies for learners to tackle.

The approach to this unit should:

- Be as practical as possible to give learners an opportunity to develop a range of customisation and integration skills.
- Make use of business scenarios to provide a context for activity.
- Where possible, engage with local employers/practitioners to complete talks and workshops to engage learn.

Delivering the learning aims

For learning aim A, while it is likely that most learners will have a relatively good understanding of different types of application software, they might not appreciate that office, Internet, security and cloud software will often need to be used together to provide a working solution to a problem.

You should begin with a recap of the different types of software and learners should be able to give examples of software for each category.

It is essential that they understand the difference between customising and integration and learners should be able to write a definition that correctly distinguishes them as approaches.

There are many potential issues with integrating applications (in particular) and learners will investigate some of the common problems from data consistency and formatting to communication protocols and issues of permissions and file sharing.

The learning aim completes with learners investigating how programming languages are used to integrate applications. Learners should consider a range of languages and should be able to demonstrate the ability to choose the right language for a particular situation.

Customisation can be achieved within a single application, such as adding automated routines to a spreadsheet or adding a menu to database software.

The final topic allows learners to develop an appreciation of how the integration of applications can be used to develop a range of solutions.

Learning aim B is practical and should begin with learners exploring the IDEs they will be using in their integration activities. You will need to produce some exercises relevant to the combination of software you have at your centre to enable the learners to practise some of the techniques they will use in their assessment. Online examples are available such as copying data from MS Access to MS Excel using Python or using Python to read an Excel file.

Learners would also benefit from using the Internet to search for websites, development forums and technical threads which will provide a useful resource to support them in their problem solving.

Testing and debugging are key to developing business solutions and while learners will have met this content before, it is unlikely that they will have designed testing strategies that will incorporate a number of different (and possibly diverse) applications or scripts working together in this way. Debugging also is essential, particularly to resolve any semantic (as opposed to running) errors.

For learning aim C, applications are customised and/or integrated to solve a business need. This is usually to meet an organisational need which could be as simple as exploiting capabilities to enhance efficiency or it could be to improve responsiveness in an enterprise context.

Learners should therefore explore a range of scenarios where integration and/or customisation would be a factor. For example,

- An automated raw materials ordering system for a manufacturing company by integrating stock control, finance and manufacturing planning.
- An environmental management system for a suite of residential building that is designed to promote energy efficiency.
- An inventory tracking system across multiple domestic appliance stores.
- A case management and problem escalation system for social workers who work out in the field.
- Absence reporting and holiday request system for a business with analytic capabilities.
- A system for tracking resources used on multiple simultaneous projects (such as man hours and other direct and indirect costs), which generates customer invoices.

- A marketing system for managing and scheduling digital marketing activity for a group of clients.
- A system for a company involved in preparing international payroll.
- An online newsfeed site that draws articles from other websites based on a user's choices.
- A secure file sharing system for a scientific collaboration.

Customisation could include:

- A series of linked spreadsheet pages managed through a customised menu using form with buttons to control access and updating to the contents and data.

Developing customised and integrated applications follows similar processes to other IT projects in that user requirements must be defined (which is often the identification and articulation of the business need), success criteria must be established, the user interface requirements agreed, and any project constraints should be identified.

Learners design a proposed solution, using traditional design tools, techniques and methods, designing test plans and test data and creating a prototype (or prototypes if applicable), as part of the design process.

The design will need to be refined, gathering feedback from others to inform changes, and documentation will need to be updated to reflect any changes or improvements to the proposed design.

Learners develop a customised and integrated solution in line with the proposed design, debugging and fixing problems as they arise to create a final version of the solution.

Functional testing is executed in line with the test plan and the solution is then compared against the user requirements to establish how well it meets the user's requirements, functions as expected and meets quality, reliability, usability and performance expectations.

Assessment model

Learning aim	Key content areas	Recommended assessment approach
A Investigate the customisation and integration of applications used by organisations to meet their needs	**A1** Types of application software **A2** Purpose of and issues with customising and integrating applications **A3** Software customisation and integration	A detailed presentation or report evaluating the customisation and integration of applications.
B Explore the technologies and techniques used to customise and integrate applications	**B1** Integrated development environments (IDEs) **B2** Programming constructs and techniques for customising and integrating applications **B3** Testing customised and integrated applications	A practical activity to develop short prototype software solutions that demonstrate the integration and customisation of different software applications. A report focusing on what went well and what did not go so well when customising and integrating software applications using different technologies.
C Develop customised and integrated applications to meet organisational needs	**C1** Design for customised and integrated applications **C2** Develop customised and integrated applications **C3** Testing customised and integrated applications	A practical activity involving defining the specification for the problem, designing and developing fully working customised and integrated software applications to meet a client's needs.

Assessment guidance

The assessment for this internally assessed unit would benefit from being divided into three assignments as shown above.

Assignment 1 should cover learning aim A. A detailed presentation or report evaluating the customisation and integration of applications.

Learners could create a slide show with supporting notes which they present to an audience, or they could present the work as a formal academic report. Learners should make use of realistic scenarios for as the basis for their presentation which should explain why integration and customisation is appropriate as a possible solution to the business problem identified in the scenario.

The organisation and business need identified should be of sufficient scope to allow learners to explore a range of possible solutions in terms of customization and integration.

What is the difference between the two approaches? How can each be used in the context of the scenario?

Learners should be able to explain any potential issues that might need to be managed.

Learners should ensure that the presentation is suitable for the audience as defined in the scenario they are given.

As an idea, the tutor could prepare two or three different scenarios that would make watching the presentations more interesting and give learners an exposure to a wider range of situations.

Assignment 2 should cover learning aim B. The assignment requires a practical activity to develop at least three short prototype software solutions that demonstrate the integration and customisation of different software applications. Learners will produce an accompanying report that discusses the prototypes and how customisation and integration have been used. The report will also focus on what went well and what did not go so well when customising and integrating software applications using different technologies.

Learners may wish to present their work as formal, professional, presentation with supporting materials which could be videoed as part of the evidence. This method would allow you to also question the learners to enable them to demonstrate deeper understanding, and hit higher grading criteria, which they may find difficult in a purely written form,

Assignment 3 should cover learning aim C. A larger project to the short prototype software solutions in assignment 2, this assignment will be a practical activity involving defining the specification for the problem, designing and developing fully working customised and integrated software applications to meet a client's needs.

You have the option for this assignment to link the activity to the scenario (or scenarios) used in assignment 1, but it is suggested that they are presented as a separate assignment.

Getting started

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

Unit 14: Customising and Integrating Applications

Introduction

Ask learners to share experiences of office productivity software they have used and discuss how they may have used it.

Ask if there was ever a time where they couldn't compete a task they needed to complete using a single piece of software. For example, collating data and presenting it for an assignment?

Pose the questions:

- Why off-the-shelf applications do not meet business needs?
- Why should businesses develop an application integration strategy?

Consider showing the short video to help answer the last question (see the link in the resources section at the end of this delivery guide).

Learning aim A: – Investigate the customisation and integration of applications used by organisations to meet their needs

A1: Types of application software

- Begin with a Q&A. How many different types of application software categories can learners identify?
- Learners should then name software products in each category.
- Carry out a straw poll – how many learners can say that they have used at least one product in each category?
- Learners carry out group work (or paired work) to investigate less well-known examples of products in the different categories. You could include the following or provide other examples:
    - Office products could include: Calligra Suite
    - Web browsers could include: Chromium (open source version of chrome) Maxthon, Lunascape Orion, Citrio and Midori
    - Security could include: Glasswire and Avira
    - Cloud software could include: Hitman Pro, Glasswire and RKill
- Learners could experiment by trying out the web browsers and comparing the functionality, response etc to the browsers they currently use like Google and Edge.
- This activity could result in a presentation to other learners who share their research and observations.

A2: Purpose of and issues with customising and integrating applications

Difference in approaches between customising and integrating applications:

- Presentation focusing on the difference between customisation and integration of applications. It should begin with a definition of each term and be supported by examples in each case. There are a number of examples in the specification that will help to inform the presentation.

Purpose of customising and integrating applications:

- Referring back to the introduction, why do businesses customise and integrate applications?
- Learners work in small groups to investigate either the Virgin or Lloyds case studies included in the resources section of this delivery guide (under News articles). What were the reasons for the integration? Were these proactive or reactive activities?

- You should lead a class discussion to explore the what, how and why in each case.
- During the discussion, you should contribute additional reasons why applications are customised or integrated (either from the list in the specification or by giving examples of your own).

Common issues with data customisation and integration:

- Presentation on the common issues with data customisation and integration – where are mistakes commonly made?
- Where possible, focus on real examples from personal experience, articles in the news and ask learners to investigate any of the issues listed in the specification such as data integration failures or automation failures (see Sainsbury's example in the resources section).

Possible solutions to common issues:

- Learners work in groups or pairs to investigate solutions to application customisation and integration issues. Allocate one of the following to each group or pair and ask learners to investigate:
    - Data security
    - Cloud technology
    - Data compartmentalisation SaaS.

A3: Software customisation and integration

- Individual activity where learners explore an allocated programming language from those in the specification list.
- Create a table that has each language listed, with columns for 'Uses' and 'Comments'. When learners feed back about their language to the class, each learner should populate the grid with the relevant information as supplied by other learners.
- Factors to consider when choosing suitable languages for use in integration and customisation projects:
- You should lead a discussion on what learners should consider when making a selection. What sort of criteria should they consider? Why? You should also explore why they might consider something less familiar, particularly if the language might provide a better solution.
- How application software can be customised:
    - Define customisation, giving examples for spreadsheets, email software, databases and a web context.
    - Create an exercise where learners carry out a simple customisation on a spreadsheet – specifically creating a macro that copies and pastes a value (or set of values) from one spreadsheet into cells in another spreadsheet.
- How application software can be integrated:
    - Define integration, explaining ERP, EDI etc as listed.
- Learners investigate the benefits of CRM systems (see the link in the resources section), either as pairs or small groups. They should write a 200-word news article that promotes the concept of a CRM and the benefits to business.

Learning aim B: Explore the technologies and techniques used to customise and integrate applications

B1: Integrated development environments (IDEs)

- Identify at least three different IDEs (e.g. Virtula Studio Code, Spyder, Atom) for learners to compare. Prepare a checklist for the functions listed in the specification and instruct learners to investigate each IDE and indicate which features or functions each one has.
- Lead class discussion – which of the IDEs offers the greatest flexibility and why?

B2: Programming constructs and techniques for customising and integrating applications

- Prepare (or acquire) a series of worksheets that will allow learners to develop and practise skills and techniques they will use to customise and integrate applications.
- Learners could use previously written code and make some adaptations, for example introduction of functions, exploring more complex parameter passing, working with different array types, for example an array of records.
- Set up a simple relational two table database (so that a query is involved), and ask learners to use a language to query data and extract the records from the database into another application (or even just into a text file).
- Create a consolidation spreadsheet by linking data between worksheets and spreadsheet files into a master file (an example exercise is included in the resources section at the end of this delivery guide).
- Learners should create simple validation routines in a range of applications.
- Provide a file containing charts and graphs linked to a data set, and then ask learners to modify the parameters and associated text, changing the chart or the graph type as part of this process.
- There are many exercises that could be carried out here – and you may be able to use exercises from other courses in your department where they exist or draw examples from programming units that can be adapted.

B3: Testing customised and integrating applications

Test customised and integrated applications:

- Lead a class discussion on testing techniques used so far in other units or assessments. Which apply for testing customised and integrated applications?
- Make a list and ask learners to define each one they understand and investigate any that they do not.
- Learners work in pairs to investigate integration testing. A link has been provided in the resources section of this guide, which will give a good starting point. How does integration testing within a programme or series of linked programmes translate to a scenario with integrated applications? Lead a class discussion.

Debugging:

- Learners should practise using debugging techniques. Syntax errors are often easy to spot. Semantic errors (errors in meaning rather than incorrect code) are less easy to spot.

Learning aim C: Develop customised and integrated applications to meet organisational needs

C1: Design for customised and integrated applications

User requirements:

- Presentation focusing on user requirements. Discuss the challenges of working with a client to define a problem and the reasons why the problem scope should be recorded once it has been

agreed (particularly to help protect against project creep). Why is it important to identify constraints, the user interface requirements and the criteria that will be used to measure success?

- Lead a discussion possible project failure due to errors in establishing user requirements. The scope creep article blames the failure of a project on three key factors. This article will be a good starting point for the discussion.

Design:

- Learners should practise designing integrated and customised applications using a range of scenarios. They should use illustrations and diagrams, flow charts to communicate processing stages, control structures and activity diagrams. The scenarios can be relatively simple, and you would benefit from having two scenarios and allocating them equally to different groups in the class. Once a group has completed their design, they should informally present the design to a group working on the other scenario. Any feedback from the observing group should be recorded. The process should then be reversed. Scenarios could include:
    - A customised interface for a group of disabled adults with poor motor skills.
    - Setting up an automated custom filter.
    - Setting up an invoicing system that draws customer data from a database.
- For each scenario worked through, learners should be able to explain their reasons for the design approach taken.

Review and refinement of the design:

- Presentation on the role of the review in the design process, posing a series of questions – for example:
    - Why is ongoing communication with the client important?
    - Why should feedback be gathered during the design stage?
    - Why should ideas and proposed solutions be refined?
    - Why should a design specification be constantly updated?
    - Why should the techniques and approach used be reviewed?

C2: Develop customised and integrating applications

Using tools and techniques:

- Using the designs created in learning aim C1, learners use a range of techniques, constructs and languages to create a prototype (this does not need to be a full prototype but should demonstrate the workings of at least part of the system).
- Carry out a review of the techniques, constructs and languages that learners used and provide feedback. Were these used correctly and efficiently? What could they have done differently?

Review and refine the solution:

- Learners should ask for feedback on their product. Do the class feel that the product met the objectives originally set out in the scenario? Are there any enhancements that could be made relatively easily to improve the functionality of the solution?

C3: Testing customised and integrated applications

Functional testing:

- The applications should be subjected to functional testing against an agreed test plan. Results should be recorded. You could consider issuing a testing log that learners could use to record the test, test data, result of the test and any actions that need to be taken (for example correcting errors – what need to be corrected and how was this achieved?).

Acceptability testing:

- The concluding part of the testing is essentially checking the final solution against a range of criteria.

- Does the solution meet all of the requirements (or does what was created fully meet the requirements it was intending to meet)?
- Does it work as expected?
- Is it reliable, usable and of good quality?

- Learners create a short presentation that explains the outcomes of the activities – what would they do differently if they did this project again?

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

This unit links to:

- Unit 2: Creating Systems to Manage Information
- Unit 4: Programming
- Unit 5: Data Modelling
- Unit 7: Mobile Apps Development
- Unit 9: IT Project Management
- Unit 11: Cyber Security and Incident Management
- Unit 15: Cloud Storage and Collaboration Tools
- Unit 18: The Internet of Things
- Unit 20: Business Process Modelling Tools.

Resources

In addition to the resources listed below, publishers are likely to produce Pearson-endorsed textbooks that support this unit of the BTEC Internationals in Information Technology. Check the Pearson website at http://qualifications.pearson.com/endorsed-resources for more information as titles achieve endorsement.

Textbooks

- Mendelsohn E, Monk E, Cook G and Brady J, *Problem Solving Cases in Microsoft Access and Excel* (14th Edition), CENGAGE Learning Custom Publishing, 2016 ISBN 9781305868625 – a highly practical and up-to-date publication that requires learners to use Excel and Access to create solutions to business problems.
- *Woo D and VanHuss SH,* Integrated Computer Applications (6th Edition), South-Western College Publishing, 2011 ISBN 9781111988098 – This book is now a little dated, but it provides useful exercises and ideas on how applications can be integrated to provide business solutions.

Videos

- *Develop an Application Integration Strategy* (4.5 minutes) – considers the state of current business IT resources and why strategically planned integration (https://www.youtube.com/watch?v=pSOMPhMCAZ4).
- *How to use Excel and Access for a solution* – this is a YouTube channel with a range of how to videos that show how Excel and Access can work together to produce business solutions (https://www.youtube.com/channel/UCPMn-fP8ruZWhE-soxNUwnA).
- *What is an API? – Fast Tech* Skills (5 minutes) – this short video explains the concept of API and how it allows access to pre-existing services (https://www.youtube.com/watch?v=B9vPoCOP7oY).
- *What is an API? (*3.5 minute) – the second short video which introduces API using different examples (https://www.youtube.com/watch?v=s7wmiS2mSXY)

Websites

- https://dev.office.com/docs – *Office Developer Centre* – providing coded examples, documentation and resources for developers seeking to customise Microsoft products.
- https://www.netguru.co/blog/software-testing – *Test all the Things!* Types and Examples of Software Tests.

- http://www.preact.co.uk/why-crm/how-businesses-benefit-from-crm – *19 Business Benefits of CRM* – 19 reasons why customer relationship management systems benefit organisations with a range of examples.
- http://web.utk.edu/~dhouston/excel/exer8.pdf – *Microsoft Excel Exercises – Linking Exercise (Exercise 8)* – creating and consolidating spreadsheets between workbooks and a master file.

News articles

http://www.campaignlive.co.uk/article/186004/virgin-relaunches-upper-class-integrated- dm-push# – *Virgin relaunches Upper Class with integrated DM push* – the integration of data, communication and a microsite to reinvigorate this Virgin product.

http://www.computerworld.com/article/2533563/it-project-management/it-s-biggest- project-failures-and-what-we-can-learn-from-them.html?page=3 – *Sainsbury's warehouse automation* – what went wrong

http://www.information-age.com/merging-lloyds-tsb-and-hboss-applications-2127203/ – *Merging Lloyds TSB and HBOS's applications* – how the organisation combined its IT applications following merger

http://www.techrepublic.com/blog/tech-of-all-trades/real-world-example-of-scope-creep/ – *Real world example of scope creep* – an anecdotal example from an IT project manager. The conclusion to this article focuses on poor communication, poor setting of project requirements and a failure to fully assess options.

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 learners to access them through the school/college intranet.