Enterprise Computing Stage 6 (Year 12) – teacher support resource

**Enterprise project**

# Teacher support resource

**Teacher note:** this resource has been designed to facilitate the ready conversion into a student booklet by removing the answers within the response windows. Teacher notes can be deleted before distributing to students.

Student name:

Class:

Teacher:

Contents

[Teacher support resource 1](#_Toc164759745)

[Unit overview 6](#_Toc164759746)

[Assessment task overview 7](#_Toc164759747)

[Submission details 8](#_Toc164759748)

[Steps to success 9](#_Toc164759749)

[Glossary 13](#_Toc164759750)

[NESA glossary keywords 15](#_Toc164759751)

[The design and production process 16](#_Toc164759752)

[Identifying and defining 17](#_Toc164759753)

[Describe the tools and processes used to manage and document the development of an enterprise system 17](#_Toc164759754)

[Problem definition 17](#_Toc164759755)

[Time and resource management including Gantt charts 17](#_Toc164759756)

[An iterative approach 17](#_Toc164759757)

[Production process and technical skills 17](#_Toc164759758)

[Testing and evaluation 18](#_Toc164759759)

[Explain the effect of the changing nature of enterprise on the development of projects 19](#_Toc164759760)

[Offshore development 19](#_Toc164759761)

[Working remotely 21](#_Toc164759762)

[Freelance work 22](#_Toc164759763)

[Enabling the growth of start-ups 22](#_Toc164759764)

[Researching and planning 24](#_Toc164759765)

[Investigate tools that support the design and development of an enterprise system 24](#_Toc164759766)

[Online collaboration 24](#_Toc164759767)

[Time/task action plans 24](#_Toc164759768)

[Process diary including ongoing evaluation 24](#_Toc164759769)

[Budget 25](#_Toc164759770)

[System flowcharts 25](#_Toc164759771)

[Data flow diagrams 27](#_Toc164759772)

[Decision trees 29](#_Toc164759773)

[Describe how computational, design and systems thinking skills are used in the design and development of an enterprise system 30](#_Toc164759774)

[Computational thinking 30](#_Toc164759775)

[Design thinking 31](#_Toc164759776)

[Systems thinking 32](#_Toc164759777)

[Integrating Computational, Design and Systems thinking 33](#_Toc164759778)

[Select key collaborating and managing criteria appropriate to the development of an enterprise project 34](#_Toc164759779)

[Designing for ease of operation and maintenance 34](#_Toc164759780)

[Clarifying each of the relevant informatics within the new system 35](#_Toc164759781)

[Outlining the role of the participants, data and components used in the system 35](#_Toc164759782)

[Negotiating user/client needs and wants 35](#_Toc164759783)

[Working collaboratively 36](#_Toc164759784)

[Producing and implementing 37](#_Toc164759785)

[Apply tools to inform the requirements and limitations of an enterprise system 37](#_Toc164759786)

[Interviews 37](#_Toc164759787)

[Surveys 38](#_Toc164759788)

[Analytical reports 39](#_Toc164759789)

[Prototypes 39](#_Toc164759790)

[Presentations of research results 40](#_Toc164759791)

[Explore and apply the most suitable development approach to develop, modify and implement an enterprise system 42](#_Toc164759792)

[Waterfall (structured) 42](#_Toc164759793)

[Agile 42](#_Toc164759794)

[Prototyping 43](#_Toc164759795)

[End-user 43](#_Toc164759796)

[Outsourcing 44](#_Toc164759797)

[Develop an implementation plan and test its feasibility for an enterprise computing system 45](#_Toc164759798)

[Design thinking 45](#_Toc164759799)

[Thinking and design tools, including storyboards, Gantt chart, decision tree 45](#_Toc164759800)

[Risk analysis 46](#_Toc164759801)

[Hardware and software integration 46](#_Toc164759802)

[Training 46](#_Toc164759803)

[Preferred system implementation method 46](#_Toc164759804)

[Methodology for testing the system 47](#_Toc164759805)

[Testing and evaluating 49](#_Toc164759806)

[Verify and validate an enterprise computing system 49](#_Toc164759807)

[Evaluating test data 49](#_Toc164759808)

[Trialling the operation and maintenance documentation 49](#_Toc164759809)

[Reviewing the impact of system implementation within relevant environments 50](#_Toc164759810)

[Modifying designs to improve functionality 50](#_Toc164759811)

[Testing, evaluating and maintaining the developed enterprise computing system 50](#_Toc164759812)

[Appendix A – project documentation template 52](#_Toc164759813)

[1. Identifying and defining 52](#_Toc164759814)

[1.1. Tools and processes for enterprise systems 52](#_Toc164759815)

[1.2. Justify tools and resources 53](#_Toc164759816)

[2. Research and planning 55](#_Toc164759817)

[2.1. Development of online collaboration tools for an enterprise system 55](#_Toc164759818)

[2.2. Collaboration and management of the enterprise project 56](#_Toc164759819)

[2.3. Systems modelling 57](#_Toc164759820)

[3. Producing and implementing 63](#_Toc164759821)

[3.1. Implementation plan 63](#_Toc164759822)

[3.2. Enterprise project 66](#_Toc164759823)

[4. Testing and evaluating 67](#_Toc164759824)

[4.1. Verification and validation 67](#_Toc164759825)

[4.2. Maintenance 68](#_Toc164759826)

[References 69](#_Toc164759827)

# Unit overview

In this unit students will develop the knowledge and skills to create a working enterprise computing system. It can be built to solve a problem or for a client’s specifications, using one of the project processes and implementation methods as per the syllabus.

The chosen project process is underpinned by the project documentation, which is based on the Enterprise Computing teaching outcomes and content areas. Students are encouraged to find clients and/or generate their own scenario for which these systems are to be built.

During Week 1 of the learning sequence, students will make contact with their client or analyse self-generated scenarios, and ascertain requirements, problem definition and system needs including any system limitations.

During Weeks 2 to 4 of the learning sequence, students will investigate possible solutions to the client’s problem or self-generated scenario, project processes and implementation methods, use systems modelling tools to analyse client requirements and develop possible solutions.

During Weeks 5 to 9 of the learning sequence, depending on the project process and implementation method chosen, students will present the possible solutions to their clients or teacher for final feedback before the full enterprise system is built.

During Week 10 of the learning sequence, students will complete the system. They will then use the problem definition and identified needs, as well as the actual system, to generate a presentation which details the features of the system, its success against the problem definition and identified needs. Students present the system to the class and/or client and submit the project documentation and system.

# Assessment task overview

**Type of task**: develop an enterprise project containing a solution, project documentation and a presentation.

**Outcomes being assessed**:

A student:

* analyses how innovative technologies have influenced enterprise computing systems   
  **EC-12-06**
* explains the social, ethical and legal implications of the application of enterprise computing systems on the individual, society and the environment **EC-12-07**
* justifies the selection and use of tools and resources to design and develop an enterprise computing system **EC-12-08**
* selects and applies methods to record the management and evaluate the development of an enterprise computing system **EC-12-09**
* evaluates the effectiveness of an enterprise computing system **EC-12-10**
* communicates an enterprise computing solution to a specific audience **EC-12-11**

[Enterprise Computing 11-12 Syllabus](https://curriculum.nsw.edu.au/syllabuses/enterprise-computing-11-12-2022) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2022.

**Suggested weighting: 30%**

Students identify a real-world problem or opportunity that can be addressed through developing an enterprise project. Students create an innovative and user-friendly system that uses appropriate technologies and tools. Students test and evaluate the system, document the project process, and present the system and project documentation to an audience simulating a client handover. The project adheres to the ethical and legal standards of enterprise computing practice.

Students:

* demonstrate skills and knowledge in Enterprise Computing by applying the principles of design thinking, data science, artificial intelligence and user experience.
* document and report the project process and outcomes using industry standard formats and styles.
* use appropriate tools and resources from enterprise computing to create a functional and user-friendly system.
* test and evaluate the system according to criteria and present the system to a specific audience using appropriate visual aids.
* consider the social, ethical, and legal implications of enterprise computing for the project.

## Submission details

Students submit:

**Component A – project documentation**

Project documentation that describes the project development in detail using clear, concise language, appropriate structure and format.

**Component B – enterprise system**

A system that allows users to interact with the enterprise project that includes data, visualisations and feedback.

**Component C – presentation**

A presentation that showcases the system features, benefits and challenges using appropriate visual aids such as video, slides, diagrams, screenshots and demonstrations.

## Steps to success

Table 1 – assessment preparation schedule

|  |  |
| --- | --- |
| Steps | What I need to do |
| Component A:  Project documentation | * Brainstorm and investigate several ideas and select one enterprise project to complete. * Complete the project documentation using the enterprise project documentation template provided that involves 4 key stages.  1. Identifying and defining 2. Research and planning 3. Producing and implementing 4. Testing and evaluating |
| 1. Identifying and defining  1.1. Tools and processes for enterprise systems  These include:   * problem definition * requirements and limitations * tools and processes. | For your enterprise project:   * describe the problem (or opportunity) and why you have selected it for this project. * analyse the problem (or opportunity) to determine the system requirements including the scale and scope. * establish criteria to evaluate the success of this project. * explain how these requirements were determined including the use of research, discussion and feedback. * outline the tools and processes required for the development of this new system. |
| 1.2. Justify tools and resources | * Identify and predict the skills required to use these tools and processes and training needed. * Justify the selection and use of tools and resources to design and develop an enterprise computing system. |
| 2. Research and planning  2.1. Development of online collaboration tools for an enterprise system. | * Explain the role of online collaboration tools as relevant to this project. * Develop a Gantt chart to see tasks and the plan on when to complete these tasks. * Develop a budget to understand and demonstrate the cost of the system creation. |
| 2.2. Collaboration and management of the enterprise project | Investigate how key criteria could positively or adversely affect this project. Including:   * designing for ease of operation and maintenance * designing for working collaboratively * allowing for negotiation of user or client needs and wants * role of informatics for example IT used in the project * role of participants, data and components in the new system. |
| 2.3. Systems modelling | * Demonstrate the use of modelling tools such as Level 0 and 1 data flow diagrams, schemas, storyboards, decision trees and system flowcharts. |
| 3. Producing and implementing  3.1. Implementation plan   * Hardware and software integration * Training * Systems implementation method * Testing methodology * Risk analysis | * Explain how the hardware and software needed for this new enterprise system will be integrated into existing hardware and software. * Select and explain the type of training most appropriate for the staff, for example, in-built software tutorials and help files, videos, in-person workshops, online tutorials and a hybrid model of training. * Select and describe how either direct, phased, parallel or pilot implementation are used in this new enterprise system. * Select and describe the role of how either functional testing, acceptance testing, live data, simulated data, beta testing and volume testing are used in the new enterprise system. * Explain the impact of cyber risks and cybersecurity breaches on the new enterprise system. |
| 3.2. Enterprise project | * Develop the enterprise project solution. * Create a process diary of the enterprise project development. * Collect screenshots of the final developed solution to annotate for the project documentation. * Consider the alignment of the system to the problem definition and tools/resources established in previous steps. * Questions to guide you include: * How well does your project align with the identified problem? * Did you select and use tools and resources effectively to develop your system? * Consider the functionality and user experience of the enterprise system. * Questions to guide you include: * Does the system function as intended? * How is the user experience when interacting with the system? |
| 4. Testing and evaluating  4.1. Verification and validation   * Evaluating test data * Training, operation and maintenance documentation | * Place the results of testing using the method identified in Section 3.1. * Evaluate the performance of the new enterprise system against the problem definition and needs of Section 1.1 and the data generated as a result of the chosen testing method. * Explain how well the training, operation and maintenance documentation (where applicable) affected the take-up of the system and alignment with the problem definition and needs of Section 1.1. |
| 4.2. Maintenance | * Explain any modifications needed based on feedback from Section 4.1. |
| Component B:  Enterprise project | * Design, where relevant, a front end and back end for the enterprise project. * Design, develop and test interface elements. * Refine the system. * Develop a working enterprise system. |
| Component C:  Presentation | * Develop a presentation which demonstrates the functionality of the system and its alignment with the requirements as listed in Section 1 of the project documentation. |
| * Explain how the system meets the project requirements and the user needs. * System features, benefits, and challenges. * System functionality. | * Develop presentation slides with screenshots of the main components of the enterprise project that directly correlate with project requirements and user needs from Section 1.1 of the project documentation. * Develop presentation slides which link the evaluation from Section 4 directly to both the system and improvements to be made in the future. * Demonstrate the system running in real-time to a live audience. * Respond to questions and comments as required. |

# Glossary

Many of the following words will gather more meaning to you as you work through this booklet.

Each time you see a new word in bold throughout this workbook you can add its definition in the table below in case you need to refer back later.

|  |  |
| --- | --- |
| Word | Definition |
| Agile | An iterative and flexible approach to software development, focusing on collaboration, customer feedback and small, rapid releases. |
| Data flow diagrams | Diagrams illustrating the flow of data within a system, showing how it is processed or transformed. |
| Decision trees | Diagrams that represent decisions and their possible consequences, often used in decision-making processes. |
| Design thinking | A process where a need or opportunity is identified and a design solution is developed. The consideration of economic, environmental and social impacts that result from designed solutions are core to design thinking. Design thinking methods can be used when trying to understand a problem, generate ideas and refine a design based on evaluation and testing. |
| End-user | The person or group who will ultimately use the system or product. |
| Enterprise system | The combination of digital systems, people and processes that collect, manage and analyse data. |
| Freelance work | Self-employed individuals offering their services to clients on a project basis, often working remotely. |
| Gantt charts | Visual tools used for project management to represent the timing of tasks or activities. |
| Hardware and software integration | Ensuring that different hardware and software components work together seamlessly within a system. |
| Iterative approach | A method of development where the project is divided into smaller parts and developed incrementally, with each iteration building upon the previous one. |
| Offshore development | Development of software or services by a company located in a different country. |
| Online collaboration | Working together on projects or tasks using internet-based tools and platforms. |
| Outsourcing | Hiring external individuals or companies to perform tasks or services instead of internal employees. |
| Prototyping | Building partial versions of a system to test ideas and gather feedback before full implementation. |
| Risk analysis | Identifying, assessing and prioritising risks to minimise their impact on a project. |
| Start-ups | Newly established businesses, typically with innovative ideas, aiming for rapid growth. |
| System flowcharts | Visual representations of the flow of data or processes within a system. |
| Verification and validation | Processes to ensure that the system meets specified requirements and functions correctly. |
| Waterfall (structured) | A sequential development approach where progress flows steadily downward through predefined phases. |
| Working remotely | Performing work from a location other than the traditional office setting, often enabled by technology and the internet. |

**Teacher note:** for students with an EALD background, the glossary can be provided complete so that they have additional time to understand the key terms using bilingual dictionaries. The glossary can be provided to students in their preferred communication mode.

# NESA glossary keywords

NESA keywords can be used in the syllabus and in the Higher School Certificate examination. Familiarisation with these keywords can assist in understanding how to write and respond to questions.

|  |  |
| --- | --- |
| Key term | Definition |
| Apply | Use, utilise, employ in a particular situation. |
| Describe | Provide characteristics and features. |
| Explain | Relate cause and effect; make the relationships between things evident; provide why and/or how. |
| Investigate | Plan, inquire into and draw conclusions about. |

[Glossary of key words](https://www.nsw.gov.au/education-and-training/nesa/hsc/student-guide/glossary) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2024.

**Teacher note:** develop, explore, select and verify are used in this topic and are not listed.

# The design and production process

Throughout your study of Enterprise Computing, you will learn about design processes and how to apply them. You will explore different types of design processes and learn how to apply them in your design project.

The design and production process:

* involves a sequence of organised steps which provide a solution to design needs and opportunities
* may take a few seconds or minutes, such as when you select what clothes to wear, or may take years as in the case with the design of a motor vehicle
* may involve one person or may involve many people
* may be simple or complex, depending on the task
* involves questioning (or evaluating) throughout the iterative process.

Figure 1 – flowchart of design and production process

Design and production process diagram.

A flowchart labelled 'Ongoing evaluation' with a two-headed arrow indicating both directions. 
The first part of the flowchart is called '1. Identifying and defining'. It says 'identify and define the needs, opportunities and wants of a computing challenge, practise the technical skills, develop evaluation criteria.' There is an arrow pointing to the next section, which is labelled '2. researching and planning'. It says 'research, generate and practise ideas, be creative and propose new approaches to problems, explore new design opportunities.' An arrow points to the next section, labelled '3. producing and implementing', it says 'build and implement ideas, apply a variety of skills and techniques to create products that meet set criteria, modify and iterate solutions'. The arrow points to the next section, labelled '4. testing and evaluating'. It says 'test and evaluate solutions/products, evaluate quality and effectiveness against the criteria, make judgements throughout the solution and use these to refine the product.'
After testing and evaluating is a big arrow called 'Review if required to improve' and it goes all the way back up to the first part of the flowchart, indicating a cycle.

# Identifying and defining

## Describe the tools and processes used to manage and document the development of an enterprise system

Managing and documenting the development of an enterprise system involves a variety of tools and processes to ensure that the project progresses smoothly and is well-documented for future reference.

By employing tools and processes, project teams can effectively manage and document the development of an enterprise system, ensuring that it meets stakeholder expectations, adheres to quality standards and delivers value to the organisation.

### Problem definition

This initial step involves clearly identifying and understanding the problem or need that the enterprise system aims to address. Techniques such as stakeholder interviews, surveys and market research may be employed to gather requirements and define project objectives.

### Time and resource management including Gantt charts

Tools like Gantt charts are used to schedule tasks and allocate resources effectively over the course of the project. Gantt charts provide a visual representation of project timelines, dependencies and resource allocations, aiding in project planning and monitoring.

### An iterative approach

Adopting an iterative development approach, such as Agile allows for continuous improvement and adaptation to changing requirements throughout the development process. Iterations are conducted in short cycles, with frequent feedback loops from stakeholders and end-users guiding further development.

### Production process and technical skills

Development tools and environments, such as integrated development environments (IDEs), version control systems, and programming languages/frameworks are utilised to build and maintain an enterprise system. Technical skills in areas such as software engineering, database management and system architecture are essential for effective development.

### Testing and evaluation

****Various testing techniques, including functional testing, acceptance testing, live data, simulated data, beta testing and volume testing, are employed to identify and rectify defects or issues in the system. Testing tools and frameworks help automate testing processes and ensure the reliability, performance and security of the enterprise system.

In preparation for Assessment task 3, complete the following steps.

1. Reflect on the previous evaluation of project solutions developed over the course of Enterprise Computing 11–12 to examine your experience and improve your final project.
2. Develop ideas through brainstorming and other creative processes. Seek out potential clients in your community or connect online.
3. Analyse the problem to gain an understanding of the requirements.
4. Record the identified requirements in your project documentation.
5. Check your skill level to utilise specific tools to plan the solution from start to finish.
6. Manage the technical requirements of the solution effectively.

Complete [Section 1.1 of Appendix A – project documentation template](#_Tools_and_processes).

## Explain the effect of the changing nature of enterprise on the development of projects

The changing nature of enterprise, influenced by factors such as technological advancements, globalisation and shifting market dynamics, has a significant impact on the development of projects, particularly those involving enterprise systems.

As enterprises evolve, they often become more complex, with diverse business processes, stakeholders and technologies. This complexity poses challenges for project development, requiring greater coordination, integration and adaptability to address evolving requirements and interdependencies.

The changing nature of enterprise demands agility, innovation, collaboration and a focus on user-centric design and security in project development. By adapting to these evolving trends and challenges, projects can effectively deliver value and support the strategic objectives of the enterprise.

### Offshore development

Offshore development is a practice that has emerged because of the changing nature of enterprise and globalisation. It involves outsourcing software development tasks and projects to companies or teams located in other countries, often with lower labour costs or specialised expertise.

Offshore development is a strategic response to the changing nature of enterprise, offering cost efficiency, access to specialised skills, extended development hours, scalability and market expansion opportunities for projects. By leveraging offshore resources and expertise, enterprises can effectively manage project development and achieve their business objectives in a globalised marketplace. The table below provides a list of the benefits of offshore development.

|  |  |
| --- | --- |
| Concept | Summary of offshore development benefit |
| Globalisation | * Facilitates expansion of operations beyond national borders. * Enterprises tap into talent pools in countries with lower labour costs or specific skills. * Access global workforce and expertise not available locally. |
| Cost efficiency | * Offshore development offers cost advantages. * Lower labour costs compared to developed countries. * Reduces expenses associated with salaries, benefits and infrastructure. |
| Access to specialised skills | * Enables access to specialised skills and expertise. * Offshore teams possess niche technical skills and industry-specific knowledge. * Helps overcome skill shortages or knowledge gaps. |
| Extended development hours | * Offshore teams operate in different time zones. * Provides round-the-clock development and support services. * Accelerates project turnaround times and increases productivity. |
| Scalability and flexibility | * Offers scalability and flexibility for enterprises. * Can quickly scale up or down development teams based on project requirements. * Minimises costs and administrative burden of hiring and managing additional staff. |
| Global talent pool | * Opens up access to a more diverse and global talent pool. * Enables recruitment of skilled professionals from different backgrounds. * Fosters innovation, creativity and problem-solving capabilities in projects. |
| Market expansion and localisation | * Facilitates market expansion and localisation efforts. * Helps enterprises enter new geographic markets or adapt products or services to local preferences. * Provides insights into local market dynamics, cultural nuances and regulatory requirements. |

### Working remotely

Working remotely is a practice that has become increasingly prevalent as a result of the changing nature of enterprise. It involves performing work-related tasks and responsibilities from a location outside of the traditional office setting, often enabled by advancements in technology and communication tools.

The development and widespread adoption of digital technologies, such as high-speed internet, cloud computing, collaboration tools and video conferencing platforms, have made remote work more feasible and accessible than ever before. These technologies enable seamless communication, collaboration and access to resources from any location with an internet connection, allowing project teams to work together effectively regardless of physical distance.

Enterprises increasingly operate on a global scale, with teams distributed across different geographic locations. Working remotely allows enterprises to leverage talent pools worldwide, tapping into diverse skill sets and expertise without the constraints of geographic proximity. Distributed teams collaborate virtually, sharing knowledge, ideas and resources to drive project development forward.

Remote work offers flexibility in terms of scheduling and location, allowing employees to balance their work responsibilities with personal commitments and preferences. This flexibility can contribute to increased job satisfaction, employee retention and overall well-being, leading to higher productivity and motivation among project team members.

For enterprises, remote work can lead to cost savings in various aspects, including reduced overhead costs associated with maintaining physical office spaces, commuting allowances and other expenses related to on-site work arrangements. By embracing remote work, enterprises can optimise resource allocation and allocate savings towards project development and innovation initiatives.

### Freelance work

Freelance work has emerged as a significant effect of the changing nature of enterprise on project development. It involves individuals offering their skills, expertise and services to clients or organisations on a temporary or project basis, rather than being employed full-time.

The modern workforce increasingly values flexibility and autonomy in their work arrangements. As enterprises adapt to this trend, they are more willing to engage freelance professionals to fulfill project-based needs rather than hiring full-time employees. Freelance work offers individuals the freedom to choose their projects, set their schedules and work remotely, aligning with the preferences of both workers and employers.

Freelance work allows enterprises to access specialised talent and expertise on-demand, without the constraints of geographic location or internal capacity. Freelancers often possess niche skills or industry-specific knowledge that may be difficult to find internally or through traditional hiring channels. By engaging freelance professionals, enterprises can augment their project teams with the necessary skills and experience to drive project success.

Freelance work offers scalability and cost efficiency for enterprises, particularly for projects with fluctuating resource needs or short-term requirements. Rather than maintaining a large in-house workforce year-round, enterprises can scale their teams up or down as needed by engaging freelancers on a project-by-project basis. This flexible approach to staffing helps optimise resource allocation and minimise overhead costs associated with full-time employees.

The dynamic nature of business environments requires enterprises to adapt quickly to changing market demands and project requirements. Freelance work enables enterprises to respond rapidly to evolving needs by engaging freelance professionals with the relevant skills and experience for specific projects. This agility in resource allocation allows enterprises to seize opportunities, overcome challenges and stay competitive in fast-paced industries.

Freelancers bring diverse perspectives, experiences and approaches to project development, fostering innovation and creativity within enterprise teams. By engaging freelancers from different backgrounds and industries, enterprises can benefit from fresh ideas, alternative viewpoints and cross-disciplinary collaboration, leading to more innovative and impactful project outcomes.

### Enabling the growth of start-ups

The changing nature of enterprise, characterised by digital transformation, agile methodologies, enhanced access to funding, supportive collaboration ecosystems and favourable regulatory policies, has created a fertile ground for the growth of start-ups. This environment not only enables start-ups to launch and scale with greater ease but also encourages innovation in project development, ultimately contributing to the dynamism and resilience of the global economy.

Many governments and economic zones have implemented policies and regulations that encourage start-up growth. These include tax incentives, grants and simplified business registration processes that make it easier for start-ups to operate and grow.

Digital technologies have lowered the barriers to entry for start-ups by reducing the need for substantial initial capital investments. Cloud computing, for example, allows start-ups to access high-powered computing resources on a pay-as-you-go basis, eliminating the need for significant upfront hardware investments.

Agile methodologies have influenced project development by emphasising flexibility, customer feedback and rapid iterations. Start-ups, which naturally gravitate towards agility due to their size and need for quick pivoting, benefit from being able to adapt and evolve their products or services swiftly in response to market feedback.

The rise of innovation ecosystems that include universities, research institutions and corporate partnerships has facilitated knowledge transfer and collaboration. These ecosystems support start-ups through access to research, talent and technology.



Complete [Section 1.2 of Appendix A: Project documentation template](#_Changing_nature_of).

# Researching and planning

## Investigate tools that support the design and development of an enterprise system

Each of these tools plays a specific role in the development of enterprise systems, from planning and design through to implementation and evaluation. Together, they form a comprehensive toolkit that supports project managers, developers and stakeholders in delivering successful enterprise computing solutions.

### Online collaboration

Online collaboration is a critical component in modern enterprise systems. Online collaboration tools enable team members to work together in real-time, regardless of their physical location. This supports the design and development of enterprise systems by facilitating communication, file sharing, project management and real-time feedback among project stakeholders.

### Time/task action plans

****Time/task action plans which are also known as Gantt charts, provide a visual representation of the project timeline, detailing when and by whom tasks are to be completed. This tool is pivotal in planning the stages of system development, assigning responsibilities, and tracking progress against deadlines. Effective use of Gantt charts ensures that project milestones are met, resources are efficiently allocated and potential bottlenecks are identified early.

Examine an example of Gantt charts in the [Enterprise Computing HSC Course Specifications](https://library.curriculum.nsw.edu.au/341419dc-8ec2-0289-7225-6db7f2d751ef/1299d565-a98e-4578-a5c6-53262a5ecc08/enterprise-computing-11-12-higher-school-certificate-course-specifications.PDF).

### Process diary including ongoing evaluation

Process diaries or logbooks are utilised to document the progress of a project, capturing regular entries that describe achievements, challenges, and reflections. This ongoing evaluation tool is vital for monitoring the development process, facilitating the identification and resolution of issues, and ensuring that the project adheres to its goals and timelines. By providing a historical record, it supports project continuity and knowledge transfer.

Examine an example of what information to put in a process diary in the [Enterprise Computing HSC Course Specifications (PDF 2.3 MB)](https://library.curriculum.nsw.edu.au/341419dc-8ec2-0289-7225-6db7f2d751ef/1299d565-a98e-4578-a5c6-53262a5ecc08/enterprise-computing-11-12-higher-school-certificate-course-specifications.PDF).

### Budget

Budget management is crucial in the development of enterprise systems. Budgets help in planning the financial resources needed for project execution, including technology, manpower and infrastructure costs. Effective budget management ensures that the project remains financially viable, prioritises spending, and can adapt to unforeseen expenses.

### System flowcharts

System flowcharts are graphical representations of the processes and devices within a system, showing how data flows and is processed. These flowcharts are essential for understanding the operational aspects of an enterprise system, facilitating the identification of inefficiencies and ensuring that all necessary processes are accounted for and optimally designed.

****As a class read [System Flowchart – A Complete Guide](https://www.zenflowchart.com/guides/system-flowchart#:~:text=-%20System%20flowcharts%20are%20the%20diagram%20type%20that,flow%20and%20how%20data%20moves%20in%20the%20flow.).

Examine [Flowchart Symbols and Notation](https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained?a=0).

Examine an example of a system flowchart in the [Enterprise Computing HSC Course Specifications (PDF 2.3 MB)](https://library.curriculum.nsw.edu.au/341419dc-8ec2-0289-7225-6db7f2d751ef/1299d565-a98e-4578-a5c6-53262a5ecc08/enterprise-computing-11-12-higher-school-certificate-course-specifications.PDF).

Creating a system flowchart involves a series of steps designed to visually represent how a system processes information and how data flows through various components of a system. It's a crucial part of system design and analysis, helping stakeholders understand the system's operations immediately.

1. **Define the purpose of the system flowchart** – understand and define why the system flowchart is needed. It could be for analysing existing systems, designing a new system or documenting processes.
2. **Scope of the system flowchart** – determine the boundary of the system to be charted. This helps in focusing the system flowchart on specific processes without unnecessary detail.
3. **Gather information on the system** – collect data on how the system operates and understand each step in the process including inputs, outputs and how each step is connected to the next.
4. **Choose symbols accurately** – familiarise yourself with standard system flowchart symbols (see the table below) and ensure you use these symbols consistently throughout the flowchart for clarity.
5. **Draft the flowchart** – map out each step in the process in sequence. Ensure that the direction of flow is clear and logical. Add additional symbols for data storage, external interfaces or telecommunications links as needed to accurately represent the system.
6. **Review and revise the system flowchart** – review the flowchart for accuracy and completeness. Ensure that it matches the actual or planned process. Share the draft with stakeholders, peers of the teacher for feedback. Different perspectives can help identify errors or areas for improvement. This might involve rearranging steps for better logic or clarity, updating labels or adding missing steps.
7. **Finalise the system flowchart** – ensure the final version is well-organised and visually clear. Validate the final flowchart against the system or process it represents to ensure it accurately reflects the workflow.

|  |  |  |
| --- | --- | --- |
| Symbol | Name | Description |
|  | Paper document | Represents the input or output of a document, specifically. Examples of an input are receiving a report, email or order. Examples of an output using a document symbol include generating a presentation, memo or letter. |
|  | Process | Represents a process, action or function. It’s the most widely used symbol in flowcharting. |
|  | Direct access storage | Represents data housed on a storage service that will likely allow for searching and filtering by users. |
|  | Online display | Represents where information will get displayed within a process flow. |
|  | Manual operation | Represents a step that must be done manually, not automatically. |
|  | Online input | Represents a step where data is entered into the system directly from an online source. |
| Telecommunications link symbol. | Telecommunications link | Represents the flow of data over networks, including local area networks (LANs), wide area networks (WANs), the internet or other communication networks. |
|  | Magnetic tape | Represents data storage or data processing that involves magnetic tape, a medium once widely used for storing large volumes of data due to its cost-effectiveness and capacity for sequential access storage. |
| Cloud symbol. | Cloud | Represents networks, especially the internet or cloud-based services, emphasising the concept of networked computing resources that are accessible remotely but whose specific inner workings and physical location are abstract or not directly relevant to the flowchart's scope. |

****Use specialised software to create a [system flowchart](https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained?a=0) for the enterprise project.

### Data flow diagrams

****Data flow diagrams (DFDs) illustrate how data moves through an information system, highlighting the inputs, outputs, storage points and routes between each operation. In the context of enterprise system development, DFDs aid in visualising the flow of information, identifying redundancies and ensuring data integrity and security. They serve as a blueprint for system architecture, supporting effective system design and development.

As a class read [What is a data flow diagram?](https://miro.com/diagramming/what-is-a-data-flow-diagram/)

The table below shows the symbols used in a data flow diagram,

|  |  |  |
| --- | --- | --- |
| Symbol name | Symbol | Description |
| Processes | Process symbol. | These circles show how data is processed within the system.  A circle represents a process. A process uses input(s) to generate output(s). |
| Data stores | Data store. | Depicted as open-ended rectangles, they indicate where data is stored.  A data store can be an electronic file or non-computer storage. |
| External entity | External entity. | Illustrated by squares, these are sources or destinations of data outside the system.  An external entity can be any person, organisation or element that provides data to the system or receives data from the system. |
| Data flow | Data flow. | A labelled, curved arrow represents the flow of data between processes, data stores and external entities. |

**DFD levels**

**Level 0 (Context diagram) –** this provides a high-level overview of the system, showing the system as a whole and its interactions with external entities.

**Level 1 DFD –** offers more detail than the context diagram by breaking down the system into major processes, showing how data flows between them and external entities. It provides a more granular look at the system's operation, but keeps the focus on the overall system rather than minute details.

**Level 2 DFD and beyond –** these diagrams dive deeper into each process depicted in a Level 1 DFD, detailing the sub-processes and their data flows. The further you go beyond Level 2, the more detailed and focused the examination of processes and data flows becomes.

When creating a DFD, start with identifying major inputs and outputs, then build a context diagram (Level 0). Expand this into a Level 1 DFD by detailing major processes and how data flows between them. You can further detail these processes in a Level 2 DFD if necessary. Always ensure your diagram is accurate and easily understandable, checking with others for comprehensibility.

****Examine examples of Level 0 and Level 1 DFDs in the [Enterprise Computing HSC Course Specifications (PDF 2.3 MB)](https://library.curriculum.nsw.edu.au/341419dc-8ec2-0289-7225-6db7f2d751ef/1299d565-a98e-4578-a5c6-53262a5ecc08/enterprise-computing-11-12-higher-school-certificate-course-specifications.PDF).

Use specialised software to create [Data Flow Diagram](https://miro.com/diagramming/data-flow-diagram/)s for the Enterprise project.

### Decision trees

A decision tree represents all possible combinations of decisions and their resulting actions. Branches are shown to describe the eventual action diagram depending on the condition at the time. Each decision path will lead to either another decision or a final action.

Decision trees represent the branching logic of decision-making processes, showing different possible outcomes based on various conditions. In designing enterprise systems, decision trees can help in planning the logic for automated decision-making components, improving system efficiency and guiding user interactions. They are particularly useful in scenarios requiring complex conditional logic and can enhance the system's usability and decision-making accuracy.

****As a class read [How to Make a Decision Tree Diagram](https://www.lucidchart.com/pages/how-to-make-a-decision-tree-diagram).

Examine examples of decision trees in the [Enterprise Computing HSC Course Specifications (PDF 2.3 MB)](https://library.curriculum.nsw.edu.au/341419dc-8ec2-0289-7225-6db7f2d751ef/1299d565-a98e-4578-a5c6-53262a5ecc08/enterprise-computing-11-12-higher-school-certificate-course-specifications.PDF).

Use specialised software to create a [decision tree](https://www.lucidchart.com/pages) for the Enterprise project.

Complete [Section 2.1 of Appendix A: Project documentation template](#_Development_of_enterprise).

Complete [Section 2.3 of Appendix A: Project documentation template](#_Systems_modelling).

## Describe how computational, design and systems thinking skills are used in the design and development of an enterprise system

The design and development of enterprise systems intricately weave together computational, design and systems thinking skills, each playing a pivotal role in creating complex, scalable and efficient solutions that meet business needs.

Combining computational, design and systems thinking skills provides a comprehensive approach to the design and development of enterprise systems. Computational thinking offers the logic and problem-solving methods needed to create effective solutions. Design thinking ensures the system is user-centred and meets real needs in intuitive ways. Systems thinking offers a big-picture perspective, ensuring the system operates efficiently within its larger environment and remains adaptable to future needs. Together, these skills enable the creation of enterprise systems that are not only technologically sound but also deeply aligned with enterprise objectives and user requirements.

### Computational thinking

Computational thinking involves solving problems in a manner that a computer could execute. It comprises decomposition, pattern recognition, abstraction and algorithmic design (see table below).

|  |  |
| --- | --- |
| Aspect | Description of computational thinking aspect |
| Decomposition | Breaking down a large system into manageable parts. In enterprise systems, this could mean segmenting the system into smaller, functional modules like inventory management, customer relations or data analytics. |
| Pattern recognition | Identifying similarities or patterns to simplify problem-solving. This is crucial in enterprise systems for recognising common user requirements or recurring system processes that can be standardised or automated. |
| Abstraction | Focusing on important information only, ignoring irrelevant detail. This is used to design database schemas or when creating APIs that interact with different system modules, ensuring that complex systems are manageable and understandable. |
| Algorithmic design | Creating a step-by-step solution to the problem or the algorithm. For enterprise systems, this could involve developing specific algorithms for data processing, optimising workflows or enhancing security measures. |

As a class discuss the following questions.

* How do you think breaking down a problem into smaller, manageable parts (decomposition) helps in the design and development of an enterprise system?
* Can you describe a scenario where pattern recognition would be crucial in optimising an enterprise system's performance?
* How might algorithms be utilised in managing data flow within an enterprise system?
* In what ways do you think abstraction simplifies the complexity of an enterprise system for users and developers?

### Design thinking

Design thinking is a user-centric approach to problem-solving that encourages businesses to focus on the people they're creating for, leading to better products, services and internal processes (see table below).

|  |  |
| --- | --- |
| Aspect | Description of design thinking aspect |
| Empathise | Understanding the user's needs within the context of the problem. For enterprise systems, this means understanding the end-users’ daily tasks, challenges and needs. |
| Define | Clearly articulating the problem. It involves defining the scope and objectives of the enterprise system based on user needs and business goals. |
| Ideate | Generating a range of ideas and solutions. This phase might involve brainstorming potential features, interfaces or architectures for the system. |
| Prototype | Turning ideas into tangible products. In enterprise systems, this could involve building a minimal viable product (MVP) or prototypes for different parts of the system to gather feedback. |
| Test | Iteratively testing the prototype with users. This step is crucial for refining the system based on real user feedback, ensuring the final product meets user needs and expectations. |

As a class discuss the following questions:

* Describe a process where empathy for the user (a key component of design thinking) could influence the features of an enterprise system. Why is understanding the user important?
* Reflect on an example where creative thinking led to an innovative feature. How can brainstorming and ideation improve the design phase of an enterprise system?
* Reflect on a situation where prototyping and testing could lead to significant improvements in an enterprise system. Why is iteration important in design thinking?

### Systems thinking

Systems thinking is a holistic approach to analysis that focuses on the way a system's constituent parts interrelate and how systems work overtime and within the context of larger systems (see table below).

|  |  |
| --- | --- |
| Aspect | Description of systems thinking aspect |
| Understanding interconnections | Recognising how different modules of the enterprise system interact and depend on each other. This is critical for ensuring smooth data flow and functionality across the system. |
| Leveraging feedback loops | Identifying and utilising feedback loops can help in refining system processes and functionalities, leading to more resilient and adaptable enterprise systems. |
| Considering the system in its environment | Understanding how the enterprise system fits within the broader business ecosystem, including its interaction with external systems, the regulatory environment and technological trends. This perspective helps in designing systems that are scalable, flexible and future-proof. |

As a class discuss the following questions:

* Explain how seeing the ‘big picture‘ (systems thinking) is crucial in the integration of different components in an enterprise system. Can you think of a system that failed because it lacked holistic planning?
* How does understanding the interdependencies within an enterprise system aid in predicting potential problems?
* In what ways can feedback loops be used to enhance the functionality of an enterprise system? Discuss the importance of feedback in maintaining system stability.

### Integrating Computational, Design and Systems thinking

As a class discuss the following questions:

* Describe a project or scenario where computational thinking, design thinking and systems thinking would need to work together to create an effective enterprise system. How do these approaches complement each other?
* Can you think of a real-world example where the integration of these thinking skills led to a successful enterprise solution? What was the role of each type of thinking in the project's success?
* Reflect on your own experience or aspirations in technology. How do you see yourself using these 3 types of thinking in your future projects or career?

## Select key collaborating and managing criteria appropriate to the development of an enterprise project

Selecting key collaborating and managing criteria is essential for the success of any enterprise project. These criteria ensure that the development process is streamlined, user-centric and efficiently managed.

Implementing these criteria requires a structured yet flexible approach to project management and development and rely on the following:

* Communication: establish clear channels and regular schedules for communication among team members and with stakeholders.
* Documentation: keep comprehensive and up-to-date documentation that is accessible to all team members.
* Feedback loops: incorporate regular feedback loops with stakeholders and users to ensure the project meets its targets and adapts to evolving needs.
* Training: provide training and resources to team members to foster a collaborative culture and ensure everyone is proficient with the tools and methodologies being used.
* Monitoring and evaluation: continuously monitor the project’s progress against benchmarks and evaluate performance to identify areas for improvement.

### Designing for ease of operation and maintenance

**Purpose**

Ensure the system is user-friendly and can be easily maintained and updated over time, reducing long-term costs and improving user satisfaction.

**Approach**

Involve end-users in the design process to gather feedback on usability. Implement modular design principles to simplify updates and maintenance. Establish clear documentation and best practices for system operation.

### Clarifying each of the relevant informatics within the new system

**Purpose**

Ensure that all data, processes and user interactions within the system are well-defined and understood by all stakeholders to prevent misunderstandings and to streamline development.

**Approach**

Develop comprehensive data dictionaries, process flowcharts and system specifications. Regularly review these informatics with stakeholders to confirm accuracy and relevance.

### Outlining the role of the participants, data and components used in the system

**Purpose**

Clearly define the responsibilities of each team member, the data flows within the system and how each component interacts with others to ensure cohesive development and operation.

**Approach**

Use role assignment matrices such as RACI (Responsible, Accountable, Consulted, Informed) to delineate team responsibilities. Utilise system and data flow diagrams to visualise component interactions and data pathways.

### Negotiating user/client needs and wants

**Purpose**

Balance what users need from the system for it to be effective with what they want for it to be desirable, ensuring a final product that is both useful and adopted by its target audience.

**Approach**

Engage in regular communication with users and stakeholders through interviews, surveys and feedback sessions. Use prioritisation techniques like MoSCoW (Must have, Should have, Could have, Won’t have this time) to manage and negotiate requirements.

### Working collaboratively

**Purpose**

Foster a team environment that encourages cooperation, shared responsibility and collective problem-solving to leverage diverse skills and perspectives for a better end product.

**Approach**

Implement collaborative tools and practices such as Agile methodologies, regular stand-up meetings and shared documentation platforms. Encourage open communication and a culture of feedback.



Complete [Section 2.2 of Appendix A: Project documentation template](#_Collaboration_and_management).

# Producing and implementing

## Apply tools to inform the requirements and limitations of an enterprise system

### Interviews

Interviews play a crucial role in gathering comprehensive insights into the requirements and limitations of an enterprise system, enabling the development team to design and deliver a solution that aligns closely with the organisation's goals and objectives.

Interviews ensure you understand the stakeholder needs. They allow direct communication with stakeholders, including end-users, managers and decision-makers, to understand their needs, pain points and expectations from the enterprise system.

Every enterprise has unique business processes and workflows. Interviews help uncover these intricacies, ensuring that the enterprise system is tailored to meet specific enterprise requirements.

Interviews help identify constraints such as budget limitations, resource constraints, regulatory requirements and technical limitations that might impact the design and implementation of the enterprise system.

The following list is a sample of interview questions.

1. Can you describe the primary objectives and goals of your enterprise system?
2. What are the current pain points or challenges within your existing system or processes that you hope this new enterprise system will address?
3. Are there any regulatory or compliance requirements that the enterprise system needs to adhere to?
4. What is the anticipated scale of operations for the enterprise system, in terms of users, data volume and transactions?
5. What integrations or interfaces will the enterprise system need to have with other existing systems or third-party applications?
6. What are the budgetary constraints or financial considerations that need to be considered for the implementation and maintenance of the enterprise system?
7. What are the security and data privacy requirements that the enterprise system must meet to ensure the protection of sensitive information?

### Surveys

Surveys are an important tool for informing the requirements and limitations of an enterprise system by:

* collecting stakeholder input
* identifying pain points and challenges
* understanding user needs and preferences
* quantifying requirements
* assessing readiness for change
* validating assumptions
* facilitating stakeholder engagement
* benchmarking performance
* providing ongoing feedback throughout the project lifecycle.

Surveys can help identify pain points, challenges and areas of dissatisfaction with existing systems or processes. By collecting feedback from users, organisations can prioritise requirements for the enterprise system that address these specific challenges and improve overall satisfaction.

Surveys provide an opportunity to validate assumptions and hypotheses about user needs and preferences. By comparing survey responses against existing data or anecdotal evidence, organisations can ensure that the requirements of the enterprise system are grounded.

Use specialised software to create a [survey](https://www.surveymonkey.com/) for the Enterprise project.

### Analytical reports

Analytical reports are essential for informing the requirements and limitations of an enterprise system by providing data-driven insights that support informed decision-making, identify areas for improvement and ensure the successful implementation and ongoing optimisation of the system.

Analytical reports can provide insights into user behaviour and preferences. By analysing user interactions with existing systems and processes, stakeholders can identify user needs and preferences that should be considered in the design of the new enterprise system.

Analytical reports can help forecast future demand and resource requirements. By analysing historical data and trends, stakeholders can anticipate changes in demand and ensure that the new enterprise system is scalable and capable of meeting future needs.

### Prototypes

Prototypes are essential for informing the requirements and limitations of an enterprise system by:

* providing a visual representation of the proposed solution
* gathering early feedback
* supporting iterative development
* testing usability and functionality
* reducing development risks
* aligning stakeholder expectations
* facilitating decision making
* managing scope creep
* supporting training and documentation efforts
* building stakeholder confidence throughout the project lifecycle.

### Presentations of research results

Presentations of research results play a crucial role in informing the requirements and limitations of an enterprise system by:

* offering a visual representation of data
* facilitating decision-making
* engaging stakeholders
* highlighting key findings
* contextualising data
* building support and alignment
* clarifying complex concepts
* addressing concerns and questions
* demonstrating progress
* driving action and next steps.

The visual representation of data, findings, and insights make complex information easier to understand and digest for stakeholders who may not be familiar with technical details.

Presentations should engage stakeholders by providing an opportunity for interactive discussions and Q&A sessions. Stakeholders can ask questions, share feedback and contribute their insights, fostering collaboration and ensuring that diverse perspectives are considered in the decision-making process. They should clarify complex concepts, technologies and methodologies used to inform the requirements and limitations of the enterprise system. Stakeholders can gain a deeper understanding of technical considerations and trade-offs, enabling more informed decision-making.

Presentations provide a platform for addressing stakeholders' concerns, questions, and objections regarding the proposed requirements and limitations of the enterprise system. By openly discussing potential issues and mitigations, presentations help build trust and confidence in the proposed solution.



Complete [Section 3.1 of Appendix A: Project documentation template](#_Implementation_plan).

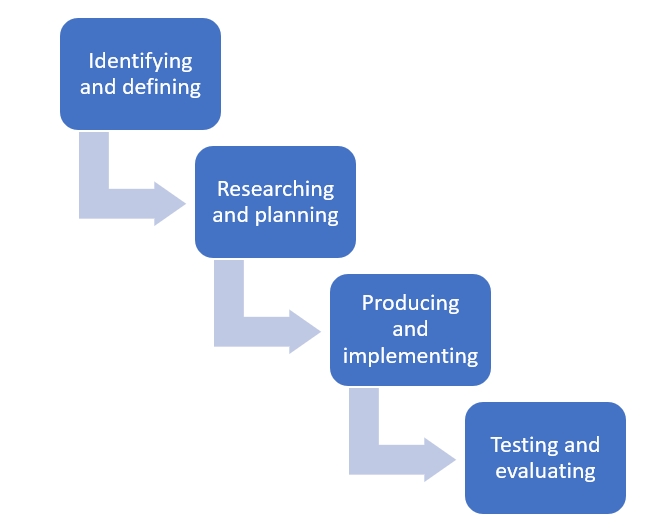
## Explore and apply the most suitable development approach to develop, modify and implement an enterprise system

### Waterfall (structured)

The waterfall development approach is a traditional software development methodology that follows a linear and sequential process with distinct phases that must be completed in a predefined order. The waterfall model is characterised by its structured and rigid nature, where progress flows downwards through each phase, like a waterfall, with no backward movement.

The waterfall development approach offers structure and predictability. It may not be suitable for projects where requirements are subject to change or where flexibility and adaptability are prioritised.

Figure 2 – the waterfall development approach using the Enterprise project production steps



### Agile

The Agile development approach is a modern iterative and incremental software development methodology that emphasises flexibility, collaboration and customer feedback. Unlike the traditional waterfall model, which follows a linear and sequential process, Agile is characterised by its adaptive and iterative nature, allowing teams to respond to changing requirements and deliver value incrementally.

The Agile approach is based on the Agile Manifesto, which values:

1. Individuals and interactions over processes and tools.
2. **Working software** over comprehensive documentation.
3. **Customer collaboration** over contract negotiation.
4. **Responding to change** over following a plan.

The Agile development approach offers a flexible and adaptive framework for building software in a collaborative and customer-centric manner. By embracing change, iteration and continuous improvement, Agile enables teams to deliver high-quality products that meet customer needs and deliver value effectively.

### Prototyping

The prototyping development approach is a software development methodology that focuses on quickly creating a working model or prototype of the intended system to gather feedback, test ideas and validate requirements before proceeding with full-scale development. Unlike traditional waterfall or Agile methodologies, which typically involve linear progressions through phases, prototyping is an iterative process where multiple prototypes may be created and refined based on user feedback and evolving requirements.

The prototyping development approach offers a flexible and iterative framework for building software that emphasises user feedback, early validation of requirements, risk reduction and innovation. By quickly creating and refining prototypes, organisations can deliver high-quality systems that meet user needs and deliver value effectively.

### End-user

The End-User Development (EUD) approach is a software development methodology that empowers non-professional developers, typically end-users or domain experts, to create and customise software solutions to meet their specific needs. Unlike traditional development approaches where software is built by professional developers, EUD shifts the responsibility of development closer to the end-users themselves. This approach acknowledges that end-users often have unique insights into their own requirements and can benefit from tools that allow them to create and modify software directly.

The End-User Development approach offers a user-centric and participatory model of software development that empowers non-professional developers to create and customise software solutions to meet their specific needs. By putting development tools directly in the hands of end-users, EUD enables greater innovation, customisation and agility in addressing business challenges.

### Outsourcing

The outsourcing development approach involves delegating software development tasks to external third-party vendors or service providers rather than handling them in-house. In this approach, organisations leverage the expertise and resources of external teams to develop, maintain and support software solutions. Outsourcing can encompass various aspects of the software development lifecycle, including design, coding, testing, deployment and ongoing maintenance.

Outsourcing allows organisations to tap into a global talent pool and access specialised skills and expertise that may not be available in-house. External vendors often have experience working on a wide range of projects and technologies, enabling them to provide valuable insights and solutions.

Outsourcing can be cost-effective compared to hiring and maintaining an in-house development team. Organisations can save on labour costs, infrastructure expenses and overhead by leveraging the resources of external vendors, particularly in regions with lower labour costs. It offers scalability and flexibility, allowing organisations to quickly scale up or down their development teams based on project requirements. External vendors can provide additional resources as needed, enabling organisations to respond to fluctuating demand without the overhead of hiring and training new employees.

The outsourcing development approach offers organisations a cost-effective, scalable and flexible solution for software development, enabling them to leverage external expertise, accelerate time-to-market, mitigate risks and focus on core competencies and innovation. However, successful outsourcing requires careful vendor selection, clear communication and effective project management to ensure alignment with organisational goals and objectives.

Complete [Section 3.1 of Appendix A: Project documentation template](#_3.1._Implementation_plan).

## Develop an implementation plan and test its feasibility for an enterprise computing system

Developing an implementation plan for an enterprise computing system involves a multi-faceted approach that ensures all aspects of the system's design, development, testing and deployment are systematically addressed.

### Design thinking

* Empathise with end-users: gather insights into the end-users' needs and challenges through surveys, interviews and observation.
* Define the problem: clearly articulate the problems the enterprise system aims to solve, aligning with business objectives and user needs.
* Ideate solutions: brainstorm possible solutions, features, and functionalities that address the defined problems.
* Prototype: develop prototypes or mock-ups of the proposed system, using tools like Figma or Sketch.
* Test: conduct user testing sessions with the prototypes to gather feedback and refine the design.

### Thinking and design tools, including storyboards, Gantt chart, decision tree

* Storyboards: create storyboards to visualise user journeys within the system. This helps in understanding the user's interaction with the system from start to finish.
* Gantt chart: use a Gantt chart to plan out the timeline of the project, including key milestones, tasks, and responsibilities.
* Decision tree: develop decision trees to outline the logic behind system processes and user decisions within the application. This aids in clarifying complex logic and decision-making pathways.

### Risk analysis

* Conduct a comprehensive risk analysis to identify potential risks associated with the project, including technical, financial, operational and market-related risks.
* Develop mitigation strategies for each identified risk, such as backup systems for data, diversifying supplier sources or implementing robust security measures.

### Hardware and software integration

* Ensure compatibility between the system's software components and the existing hardware infrastructure. This might involve hardware upgrades or modifications.
* Plan for the integration of new software with existing enterprise systems, ensuring seamless data exchange and interoperability.

### Training

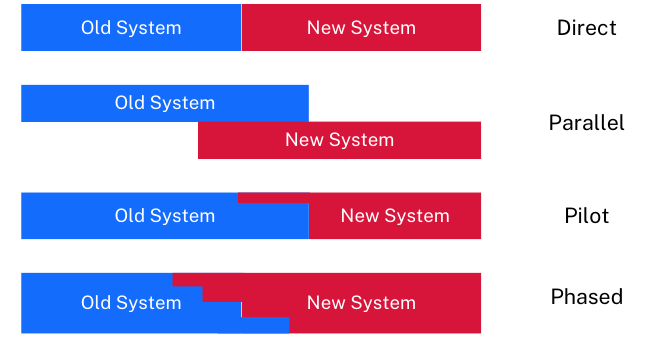
* Develop a training plan for end-users and IT staff that covers the operation, management, and troubleshooting of the new system.
* Consider different training formats, such as workshops, e-learning modules and hands-on training sessions to accommodate diverse learning preferences.

### Preferred system implementation method

Each of these implementation methods has its advantages and disadvantages and the choice depends on the specific context of the project, including its size, complexity and the organisation's risk tolerance.

* **Direct implementation** involves switching from the old system to the new system in one single action. All users move to the new system on a set date, and the old system is retired immediately. This method is straightforward and quick but can be risky if the new system has not been thoroughly tested.
* **Parallel implementation**: in parallel implementation, both the old and new systems run simultaneously for a period. This allows for a comparison between the 2 systems to ensure that the new system operates correctly before the old system is decommissioned. While safer than direct implementation, running 2 systems in parallel can be resource-intensive.
* **Pilot implementation** involves rolling out the new system to a small, manageable group of users before a full-scale implementation. This method allows organisations to identify any issues or necessary adjustments in a controlled environment, reducing the risk of widespread problems.
* **Phased implementation** involves gradually implementing the new system in phases or modules. Each phase is rolled out and stabilised before moving on to the next, allowing for incremental testing, training and adaptation. This method can reduce risk and disruption but may take longer to fully implement the new system.

Figure 3 – the 4 implementation methods





Complete [Section 3.1 of Appendix A: Project documentation template](#_3.1._Implementation_plan).

### Methodology for testing the system

Each testing method plays a critical role in the software development lifecycle, helping to identify and rectify issues before the system goes live. Understanding these methods is essential for ensuring systems undergo testing prior to implementation.

* **Functional testing**: evaluates a system's compliance with its specified requirements. It focuses on testing the functionalities of the software system, including user commands, data manipulation, searches, business processes, user screens, and integrations. The main objective is to ensure that the system performs its intended functions correctly.
* **Acceptance testing**: is the final phase of testing before the system is handed over to the end-users. It is performed to ensure that the system meets all agreed-upon specifications and requirements and is ready for operational use. Acceptance testing can be conducted by the end-users, clients, or stakeholders.
* **Live data**: involves using actual data in the system's operational environment to test its functionality and performance. This method is useful for validating the system's ability to handle real-life scenarios and data volumes. It helps identify issues that may not be evident with test data, such as data format issues, integration problems with other systems, and performance under normal operational loads.
* **Simulated data**: uses artificial or generated data that mimics real operational data to test the system. This method allows testers to model various scenarios, including edge cases and conditions that are difficult to replicate with live data. It is particularly useful in the early stages of development when actual operational data may not be available or in scenarios where using real data could pose security or privacy risks.
* **Beta testing**: is a type of user acceptance testing that involves releasing the software to a limited audience outside of the development team but within the real user environment. This group of end-users uses the system in real conditions and provides feedback on its functionality, usability, and any bugs or issues. Beta testing helps ensure the system meets user needs and expectations before a full-scale launch.
* **Volume testing** is part of performance testing and assesses the system's performance under varying database volumes. It aims to determine how the system behaves when handling a large amount of data, focusing on response times, data processing speeds and system stability. This testing is crucial for systems expected to process high volumes of data, ensuring they can maintain performance without degradation.



Complete [Section 3.1 of Appendix A: Project documentation template](#_3.1._Implementation_plan).

# Testing and evaluating

## Verify and validate an enterprise computing system

Verifying and validating an enterprise computing system are critical phases that ensure the system meets its intended requirements and operates correctly in its target environment. These processes involve a combination of technical reviews, testing and real-world operational scenarios.

By thoroughly applying these verification and validation steps, organisations can ensure that their enterprise computing systems are reliable, secure and aligned with business objectives, providing a solid foundation for operational success.

### Evaluating test data

**Purpose**

To ensure that the system processes data accurately and as expected across a range of scenarios, including edge cases.

**Approach**

Create a comprehensive test suite with a wide range of test data that covers all possible inputs the system might encounter. Use automated testing tools where possible to increase efficiency and coverage. Analyse test results to identify any discrepancies or failures.

### Trialling the operation and maintenance documentation

**Purpose**

To confirm that the documentation provided is sufficient for operating and maintaining the system without direct support from the development team.

**Approach**

Conduct hands-on trials of the system using only the provided documentation. This can involve staff training sessions or simulation exercises. Feedback from these sessions should be used to identify any areas where the documentation is lacking or unclear and require revisions.

### Reviewing the impact of system implementation within relevant environments

**Purpose**

To understand how the system affects its operational environment, including any unintended side effects or disruptions to existing processes.

**Approach**

Implement the system in a controlled, real-world environment similar to the production environment. Monitor system performance and gather feedback from users and stakeholders on its impact. Evaluate the system's integration with existing infrastructure and any changes in operational efficiency or productivity.

### Modifying designs to improve functionality

**Purpose**

To refine the system based on feedback and test results, ensuring optimal performance and user satisfaction.

**Approach**

Use an iterative design process where feedback from testing and trial runs is used to make informed design improvements. Prioritise changes based on their impact on system functionality and user experience. Ensure that modifications go through the same rigorous testing and validation process as the original design.

### Testing, evaluating and maintaining the developed enterprise computing system

**Purpose**

To ensure the system remains effective and efficient throughout its operational life, adapting to changes in technology, business processes and user needs.

**Approach**

Establish a routine maintenance schedule that includes regular updates, security patches and performance reviews. Implement continuous integration and deployment (CI/CD) pipelines for efficient testing and deployment of changes. Use metrics and feedback to continually evaluate the system's performance and user satisfaction, making adjustments as necessary.



Complete [Section 4 of Appendix A: Project documentation template](#_Testing_and_evaluating).

# Appendix A – project documentation template

## Identifying and defining

Guiding steps for your enterprise project.

1. Describe the problem (or opportunity) and why you have selected it for this project.
2. Analyse the problem (or opportunity) to determine the system requirements including the scale and scope.
3. Establish criteria to evaluate the success of this project.
4. Explain how these requirements were determined including the use of research, discussion and feedback.
5. Outline the tools and processes required for the development of this new system.

### Tools and processes for enterprise systems

**Problem definition**

Students **analyse** the problem by **describing** each of its individual components and **explaining** how each of these components contribute to the problem needing resolution.

**Ascertaining requirements and limitations**

Students **explain** methods used in this project to ascertain requirements and limitations of the problem definition above from this list given below:

* Interviews
* Surveys
* Analytical reports
* Prototypes
* Presentation of research results.

**Tools and processes**

Using the table below, students **describe** how tools and processes can or can’t be used to develop the new system based on the problem definition above.

|  |  |
| --- | --- |
| Tool or process | Description |
| Time and resource management including Gantt charts |  |
| Production process and technical skills |  |
| Testing and evaluation |  |
|  |  |
|  |  |

Production process refers to development approach which may be iterative, waterfall (structured), agile, prototyping, end-user or outsourcing.

### Justify tools and resources

**Explanation of the changing nature of enterprise on projects**

Using the table below, students **explain** each of these ways of working and their effect on the enterprise project.

|  |  |
| --- | --- |
| Ways of working | Explanation of the effect on the enterprise project |
| Offshore development |  |
| Working remotely |  |
| Freelance work |  |

After completion of Section 1 use the following template to reflect upon your responses.

You can utilise a peer or the teacher to complete this checklist.

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Developing | Developed | Highly developed |
| The response answers the question. |  |  |  |
| The idea being written about is clear. |  |  |  |
| The writing has a logical structure. It makes sense when you read it through. |  |  |  |
| There is a clear reference to syllabus content. |  |  |  |
| There is topic specific vocabulary in the response. |  |  |  |
| Sentences make sense. |  |  |  |
| Capital letters, full stops and other punctuation is used. |  |  |  |
| Accurate spelling of challenging words. |  |  |  |

## Research and planning

### Development of online collaboration tools for an enterprise system

**Online collaboration**

Students **explain** how online collaboration tools are used in this project.

**Time/task action plans**

Students **develop** a Gantt chart that details the tasks required to be completed, resources required and person or people assigned to each task. Students ensure that the timeline does not exceed the project due date.

**Budget**

Students **develop** a budget for how much this project would cost to implement. Items to consider include but are not limited to:

* hardware
* software
* personnel
* additional costs such as electricity and internet access.

### Collaboration and management of the enterprise project

**Key criteria for enterprise project development**

Students **explain** how these key criteria given below are or are not applicable to this enterprise project.

|  |  |
| --- | --- |
| Key criteria | Explanation |
| Designed for ease of operation and maintenance |  |
| Designed for working collaboratively |  |
| Allows for negotiation of user/client needs and wants |  |
| The role of each software used to create the system |  |
| The role of the participants, data and components are outlined clearly |  |

### Systems modelling

Students **develop** the tables and diagrams in the space provided. Students should consult the [Enterprise Computing Course Specifications (PDF 2.3 MB)](https://library.curriculum.nsw.edu.au/341419dc-8ec2-0289-7225-6db7f2d751ef/1299d565-a98e-4578-a5c6-53262a5ecc08/enterprise-computing-11-12-higher-school-certificate-course-specifications.PDF) guide should they require further detail, exemplars or information. Each subsection below should be completed with Section 1.1 in mind.

**Data flow diagrams**

Students **develop** data flow diagrams (DFDs) at Level 0 and Level 1. These diagrams should explicitly include the variables from the data dictionaries previously identified as well as the needs identified in Section 1.1.

*Level 0 Data flow diagram*

*Level 1 Data flow diagram*

**Schema**

Students **develop** schemas for relational database solutions. Remove this section if not applicable to your enterprise system.

**Storyboards**

Students **develop** storyboards, visually representing the software solutions they will build.

**Decision trees**

Students **develop** decision trees to visually outline the logic flow and chain of decisions or selections the final solution will need.

**System flowcharts**

Students **develop** system flowcharts that both meet the problem definitions and project needs as identified in Section 1.1 as well as incorporate the information from the other diagrams and charts in this section.

## Producing and implementing

### Implementation plan

Students develop an implementation plan that details the following components and **explains** their relevance to the project.

**Hardware and software integration**

Students **describe** in the following table how they will integrate (install) required hardware and software into the enterprise/organisation.

|  |  |
| --- | --- |
| Hardware and software | Integration into enterprise |
|  |  |
|  |  |
|  |  |

**Training**

Students **describe** how they will train the staff to use the new enterprise solution. Methods may include:

* in-built software tutorials and help files.
* in-person workshops and classes.
* online tutorials via videos.
* online tutorials via video conferencing software such as Microsoft Teams or Zoom.
* hybrid model mixing online tutorials and in-person workshops and classes.

**Preferred systems implementation method**

Students **describe** how the systems implementation method will be used in this enterprise project. Systems implementation methods include:

* Direct
* Phased
* Parallel
* Pilot.

**Testing methodology**

Students **describe** how they will test their enterprise project. Testing methodologies include:

* functional testing
* acceptance testing
* live data
* simulated data
* beta testing
* volume testing.

**Risk analysis**

Students **explain** in the following table how risks associated with the introduction of this new enterprise system would affect individuals within the organisation and the organisation itself.

|  |  |
| --- | --- |
| Risk factor | Effect on individuals and organisation |
| Cyber risks |  |
| Internal vulnerabilities |  |
| External vulnerabilities |  |
| Likelihood of exploitation |  |
| Cybersecurity breaches |  |
| Financial loss |  |
| Reputational damage |  |
| Operational disruptions |  |
| Legal and regulatory ramifications |  |

### Enterprise project

Students complete process diary entries outlining the system development in the following table.

|  |  |
| --- | --- |
| Date: | Process diary entry |
| Person making entry |  |
| Progress since the last entry |  |
| Tasks achieved |  |
| Stumbling blocks or issues encountered and how they were managed |  |
| Possible approaches for upcoming tasks |  |
| Reflective comments |  |
| Resources used |  |

Students include screenshots of their final developed solution here as part of a project development log. Each screenshot should include a caption that explains how it links to the:

* problem definition and needs identified in Section 1.1.
* components of Section 2.3 such as the storyboards, data dictionaries and so on.

## Testing and evaluating

### Verification and validation

**Evaluating test data**

Students place the results of their testing below, based upon the testing method as identified in Section 3.1. In the table below, students **evaluate** the effectiveness of the developed enterprise system in relation to the problem definition and needs in Section 1.1.

**Test results**

**Analysis and evaluation**

|  |  |  |
| --- | --- | --- |
| Problem definition and needs | Achieved? (Y/N) | Evaluation against test results |
|  |  |  |
|  |  |  |
|  |  |  |

**Training, operation and maintenance documentation**

Students **explain** the effectiveness of the training method chosen at Section 3.1 based upon user feedback, Section 1.1 and the evaluation of the test data.

### Maintenance

**Modification of enterprise system**

Students **explain** modifications, and the need for them, to the enterprise system built based upon feedback and analysis of test data from Section 4.1.

**Note:** each modification should have its own paragraph.

# References

This resource contains NSW Curriculum and syllabus content. The NSW Curriculum is developed by the NSW Education Standards Authority. This content is prepared by NESA for and on behalf of the Crown in right of the State of New South Wales. The material is protected by Crown copyright.

Please refer to the NESA Copyright Disclaimer for more information <https://educationstandards.nsw.edu.au/wps/portal/nesa/mini-footer/copyright>.

NESA holds the only official and up-to-date versions of the NSW Curriculum and syllabus documents. Please visit the NSW Education Standards Authority (NESA) website <https://educationstandards.nsw.edu.au> and the NSW Curriculum website <https://curriculum.nsw.edu.au>.

[Enterprise Computing 11–12 Syllabus](https://curriculum.nsw.edu.au/learning-areas/tas/enterprise-computing-11-12-2022/overview) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2022.

[Enterprise Computing 11–12 Course Specifications](https://library.curriculum.nsw.edu.au/341419dc-8ec2-0289-7225-6db7f2d751ef/1299d565-a98e-4578-a5c6-53262a5ecc08/enterprise-computing-11-12-higher-school-certificate-course-specifications.PDF) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2022.

Lucid Software Inc. (2024)[*Decision tree*](https://www.lucidchart.com/pages), Lucidchart website, accessed 3 April 2024.

Lucid Software Inc. (2024) [*Flowchart Symbols and Notation*](https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained?a=0), Lucidchart website, accessed 3 April 2024.

Lucid Software Inc. (2024)[*How to Make a Decision Tree Diagram*](https://www.lucidchart.com/pages/how-to-make-a-decision-tree-diagram), Lucidchart website, accessed 3 April 2024.

Miro (2024)[*Data Flow Diagram*](https://miro.com/diagramming/data-flow-diagram/), Miro website, accessed 3 April 2024.

Miro (2024)[*What is a data flow diagram?*](https://miro.com/diagramming/what-is-a-data-flow-diagram/), Miro website, accessed 3 April 2024.

NESA (NSW Education Standards Authority) (2021) ‘[Glossary of key words](https://www.nsw.gov.au/education-and-training/nesa/hsc/student-guide/glossary)’, Student guide, NESA website, accessed 10 April 2024.

Zen Flowchart (n.d.) [*System Flowchart - A Complete Guide*](https://www.zenflowchart.com/guides/system-flowchart#:~:text=-%20System%20flowcharts%20are%20the%20diagram%20type%20that,flow%20and%20how%20data%20moves%20in%20the%20flow.), Zen Flowchart website, accessed 3 April 2024.

**© State of New South Wales (Department of Education), 2024**

The copyright material published in this resource is subject to the Copyright Act 1968 (Cth) and is owned by the NSW Department of Education or, where indicated, by a party other than the NSW Department of Education (third-party material).

Copyright material available in this resource and owned by the NSW Department of Education is licensed under a [Creative Commons Attribution 4.0 International (CC BY 4.0) license](https://creativecommons.org/licenses/by/4.0/).

[](https://creativecommons.org/licenses/by/4.0/)

This license allows you to share and adapt the material for any purpose, even commercially.

Attribution should be given to © State of New South Wales (Department of Education), 2024.

Material in this resource not available under a Creative Commons license:

* the NSW Department of Education logo, other logos and trademark-protected material
* material owned by a third party that has been reproduced with permission. You will need to obtain permission from the third party to reuse its material.

**Links to third-party material and websites**

Please note that the provided (reading/viewing material/list/links/texts) are a suggestion only and implies no endorsement, by the New South Wales Department of Education, of any author, publisher, or book title. School principals and teachers are best placed to assess the suitability of resources that would complement the curriculum and reflect the needs and interests of their students.

If you use the links provided in this document to access a third-party's website, you acknowledge that the terms of use, including licence terms set out on the third-party's website apply to the use which may be made of the materials on that third-party website or where permitted by the Copyright Act 1968 (Cth). The department accepts no responsibility for content on third-party websites.