

Deadline 04th  
September 2019

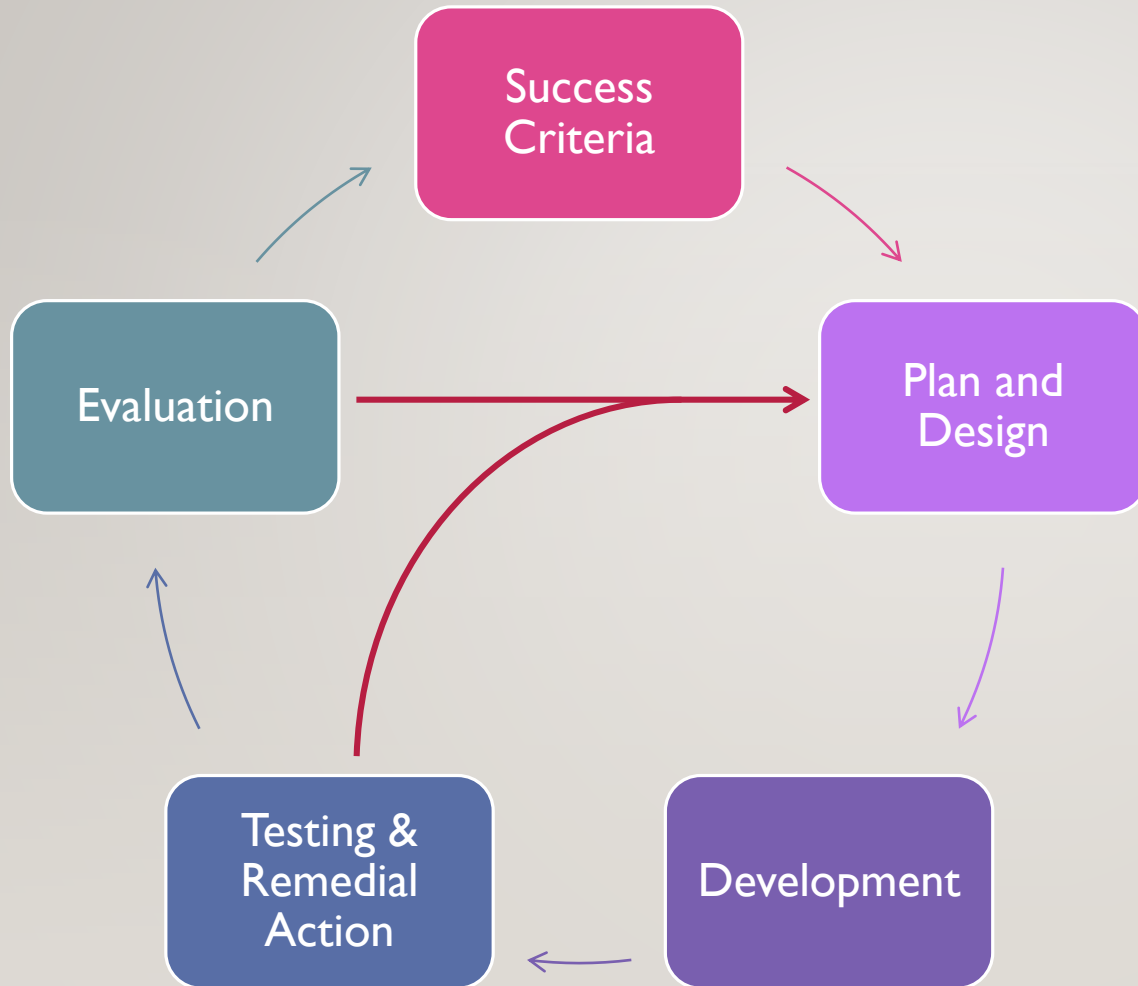
# YEAR 13

---

COMPUTER SCIENCE BRIDGING WORK

- 
- By now you should have completed the analysis and design of your project.
  - Don't worry if it is not perfect as you will get a chance to improve it in September.
  - The most important thing is to complete the design and most of the development of your project.

# Iterative Development Process



## Success criteria

- What will a successful solution do

## Planning and design

- Pseudocode, flowcharts, DFDs, Class Definitions/UML etc

## Development

- Narrative of steps taken
- Annotated Code
- Discussion of methodology

## Testing and remedial actions

- narrative of changes made

## Evaluation

- link to success criteria
- evidence of success or not

# Design – brief overview

## 13–15 marks

- Broken the problem down systematically into a series of smaller problems suitable for computational solutions, explaining and justifying the process.
- Defined in detail the structure of the solution to be developed.
- Described the solution fully using appropriate and accurate algorithms justifying how these algorithms form a complete solution to the problem.
- Described, justifying choices made, the usability features to be included in the solution.
- Identified and justified the key variables / data structures / classes (as appropriate to the proposed solution) justifying and explaining any necessary validation.
- Identified and justified the test data to be used during the iterative development of the solution.
- Identified and justified any further data to be used in the post development phase.

- You should be using a suitable design methodology for your chosen project + language. For instance, use of class diagrams for OOP, or Top Down Design for Procedural.
- Each smaller problem should be justified as to why this is a suitable “chunk” and how it fits in to the grand scheme.
- Flow Charts, Data Flow, Pseudocode etc should be used appropriately to then design the program.
- There is no specified design methodology.
- Psuedocode is inherently required
- The emphasis is that a 3<sup>rd</sup> party should be able to implement the solution directly from the designs
- Suitable data structure design should be used as well
- Test data for the initial designs should be specified on a ‘module by module’ approach, as well as test data that will be used to ensure all of the Success Criteria are met
- Awarding of marks for Design and Test plans may be awarded at a later date if the candidate decides that initial designs/tests may need modification/adaptation

# DEVELOPMENT

---

- This is the main part of what you will be doing over the summer.
- It is very important that you complete the development of your project in its entirety.
- As you complete different sections of it, you should take regular print screens as evidence of the iterative development.
- You should also take print screen evidence of the testing of your product.
- Don't worry about annotating it in too much detail, just ensure that you have sufficient evidence and that the product is completed by 3<sup>rd</sup> September.

# Development – brief overview

13–15 marks

- Provided evidence of each stage of the iterative development process for a coded solution relating this to the break down of the problem from the analysis stage and explaining what they did and justifying why.
- Provided evidence of prototype versions of their solution for each stage of the process.
- The solution will be well structured and modular in nature.
- Code will be annotated to aid future maintenance of the system.
- All variables and structures will be appropriately named.
- There will be evidence of validation for all key elements of the solution.
- The development will show review at all key stages in the process.

- Avoid ‘death by screen shot’
- Prints of Code blocks/modules with descriptions are fine – no need for ‘line by line account’
- Highlight unique features or ‘complex coding’ etc – i.e. Showcase the complexity
- Develop in blocks/chunks as defined in the design
- Test as you go – no need to wait until the end
- Agile/Iterative/RAD styled development
- Good annotation in code will reduce quantity of writing
- ‘Lean and Mean’ – keep the development as concise as possible
- Ensure good naming conventions are used (should be included in designs). Candidates that use Form\_1 etc, are not showing good maintainability of code.
- Highlight the validation – emphasise robust coding
- Check Points/Milestones after each module – review back to Success Criteria
- Signed of development by Stakeholders for each iteration

# Testing (Development) – A Brief overview

## 9–10 marks

- Provided evidence of testing at each stage of the iterative development process.
- Provided evidence of any failed tests and the remedial actions taken with full justification for any actions taken.

- Test at the end of each ‘module’
- No need to screenshot every test – especially if repetitive
- Highlight failed tests
- Must be enough to convince moderator of working solution
- It is fine to re-design etc at this point and you can still refer to this redesign to reflect a full working solution when awarding marks in the Design Section
- Any modification of test plans may also be used in the same way
- Must show re-development and testing where appropriate to gain full marks
- Ensure all white box testing (functionality) is completed at this stage.
- Testing for Use Acceptance (black box) is contained as part of evaluation
- This may form part of the testing they do as they develop, but is awarded in AO3.3, rather than AO3.2