



# **BCS Higher Education Qualifications**

## **Diploma in IT**

# **Software Engineering Syllabus**

Version 3.1

June 2019

This qualification is regulated by one or more of the following: Ofqual, Qualifications Wales, CCEA Regulation or SQA.

# Contents

---

1. Change History	3
2. Rationale	3
3. Aims	3
4. Objectives	3
5. Prior Knowledge Expected	3
6. Format and Duration of the Examination	4
7. Syllabus Detail	4
8. Recommended Reading List	5
9. Contact Points	5

## 1. Change History

Any changes made to the syllabus shall be clearly documented with a change history log. This shall include the latest version number, date of the amendment and the changes made. The purpose is to identify quickly what changes have been made.

Version Number	Date	Changes Made
Version 1.0		Released
Version 2.0	June 2016	Re-formatted with syllabus numbering – no change to content
Version 3.0	Dec 2016	Regulated statement added.
Version 3.1	Feb 2019	Updated reading list and minor update to content 1, 3 and 5

## 2. Rationale

This module is for those who wish to understand an introduction to Software Engineering and the skills necessary to create software products and applications that are effective, maintainable and value-for-money.

## 3. Aims

- To apply the knowledge of a disciplined approach to the development of software and to the management of the software product lifecycle

## 4. Objectives

Upon successful completion of this module, candidates will be able to demonstrate their competence in, and their ability to:

- Explain the background of the software crisis and the need for an engineering approach
- Appreciate the distinction between software programming and an engineering approach to the development of a software product
- Create models of software data and processes using object oriented modelling approaches such as the UML
- Describe and evaluate software tools and technology to enhance productivity and quality of software development
- Demonstrate skills of software documentation, quality assurance and evaluation, and testing as part of software development
- Describe development contexts and can apply estimation methods for planning these contexts

## 5. Prior Knowledge Expected

Candidates must have achieved the Certificate in IT or have an appropriate exemption to be entered for the Diploma in IT. Candidates are expected to have used a variety of programming languages and to be familiar with a range of software development technologies.

Candidates are required to become a member of BCS, The Chartered Institute for IT to sit and be awarded the qualifications. Candidates may apply for a four-year student membership that will support them throughout their studies.

## 6. Format and Duration of the Examination

The examination is a two-hour closed book examination (no materials can be taken into the examination room) based on the syllabus in this document.

Examinations are held twice a year and are undertaken in normal examination conditions with one or more duly appointed invigilators.

The pass mark is 40%.

## 7. Syllabus Detail

Category	Ref	Content
1 SOFTWARE ENGINEERING	1.1	The nature of software
	1.2	Theoretical models
	1.3	Motivation for the development of the Software Engineering discipline
	1.4	The cost of maintenance
	1.5	The cost of quality
2 SOFTWARE ENGINEERING KEY PRACTICES	2.1	The multidisciplinary nature of software design
	2.2	Team work
	2.3	Productivity
	2.4	Testing
	2.5	Product maintenance
	2.6	Software product life cycle
3 SOFTWARE DEVELOPMENT MODELS AND METHODS	3.1	Design principles (transparency, separation of concerns, abstraction, modularity and development by incremental methods)
	3.2	Notations for describing software components and architecture
	3.3	UML modelling of use cases for a logical/end-user view, system components and architecture for the development view, behaviour and deployment for process and physical implementation views
4 VALIDATION, VERIFICATION, AND TESTING	4.1	Product and process visibility
	4.2	Traceability in software systems and processes
5. SOFTWARE ENGINEERING TOOLS AND ENVIRONMENTS	5.1	Computer Aided Software Engineering (CASE) tools
	5.2	Role of repositories in supporting incremental development
	5.3	Software reuse and evolution
6. PROJECT MANAGEMENT	6.1	Project estimating and project planning
	6.2	Management and maintenance of software products in the consumer marketplace
	6.3	Total cost of system ownership
	6.4	Software life-cycle cost modelling
	6.5	Project development cost modelling
	6.6	Project and product risk management

## 8. Recommended Reading List

Software Engineering 1	ISBN 10	ISBN 13
<b>Primary Texts</b>		
• Sommerville, I., Software Engineering, Pearson, (10th Ed), 2015	1292096131	978-1292096131
• Pressman, R. S., Maxim, B., Software Engineering: A Practitioner's Approach, McGraw-Hill Education, (8th Ed), 2014	1259253157	978-1259253157
<b>Other Texts</b>		
• Brooks, F. P., The Mythical Man-Month, Addison-Wesley, 1995	0201835959	978-0201835953
• Martin, R. C., Clean Architecture, Prentice Hall, 2017	0134494164	978- 0134494164
• Stephens, R., Beginning Software Engineering, Sybex, 2015	8126555376	978- 8126555376
• Wysocki, R. K., Effective Project Management: Traditional, Agile, Extreme, Wiley India, (7th Ed), 2014	8126552207	978-8126552207
• Demarco T., Lister T., Peopleware: Productive Projects and Teams, Addison Wesley, (3rd Ed), 2016	0321934113	978- 0321934113

## 9. Contact Points

### Email:

Customer Service team via <https://www.bcs.org/contact-us>

### Phone:

UK: 01793 417417

Overseas: +44 (0)1793 417417

Lines are open Monday to Friday, 08.15 a.m. to 5.15 p.m. UK time.

### Website:

[www.bcs.org/heg](http://www.bcs.org/heg)

### Post:

BCS, The Chartered Institute for IT  
First Floor, Block D, North Star House, North Star Avenue,  
Swindon SN2 1FA, United Kingdom