

BCS Certificate in Systems Development Essentials

Learn via: **Classroom / Virtual Classroom / Online**

Duration: **3 Gün**

Overview

BCS müfredatını esas alan bu eğitim, sistem geliştirmenin temel becerileriyle ilgilidir. Sistem geliştirmenin tüm unsurlarının temelini oluşturduğuna inanıldığından sistem yaşam döngüsüne, sistem araştırmasına, modellemeye ve kalite güvenceye odaklanmaktadır. Eğitim ayrıca katılımcılara sistem geliştirme çalışmalarının nasıl düzenleneceğini de tanıtılmaktadır. BCS müfredatı, farklı genel yaşam döngüsü türleri, yöntemleri ve yaklaşımlarını birbirinden ayırdığı için özel bir yaklaşıma daha ayrıntılı olarak eğilmeyi gerektirir.

İnsanlar gerek işletme gerekse de teknik bir arka plandan uygulama sistemleri analizine geçtiklerinden uygulama sistemleri geliştirme sürecini ve sistem analistlerinin, çözüm mimarlarının da tasarımcıların rollerini daha temel bir şekilde anlamaları gerekmektedir. Bu eğitim de bu temel noktaları sunmakta ve başarılı sistem geliştirme projelerinin tümünde önemli bir rol oynadıklarından sistem araştırması ve kalite güvence tekniklerine odaklanmaktadır.

Ayrıca sistem geliştirme çalışmalarının nasıl düzenlenebileceğini de incelemekte ve genel yaşam döngüsü türleri, yöntemleri ve yaklaşımları arasındaki ayrımı ortaya koymaktadır. Eğitimin sonunda katılımcılar, Sistem Geliştirmenin Temel Unsurları alanında BCS Sertifikası elde edebilmek için bir sınava girebileceklerdir.

Bu eğitim, Çözüm Geliştirme alanında BCS Diploması almaya yönelik QA programının bir parçası olarak alınabilir. Bu eğitim, kapsadığı analiz becerilerinin uygulamalı olarak da ele alındığı, üç günlük yoğun bir eğitimidir. Katılımcılar hem alıştırmalara hem de örnek vaka görevlerine katılmaktadırlar. BCS sınavına katılan katılımcıların ise hem eğitim konularını hem de örnek sınav sorularını gözden geçirmek için her akşam yaklaşık 60-90 dakikalarını ayırmaları gerekmektedir.

BCS Sistem Geliştirme Diplomasına giden QA programını tamamlamak isteyen katılımcıların almaları önerilen eğitimlerin sıralaması ise şu şekildedir:

1. Sistem Geliştirmenin Temel Unsurları,
2. Sistem Modelleme Teknikleri (UML) veya Sistem Modelleme Teknikleri (Yapılandırılmış),
3. Sistem Tasarım Teknikleri,
4. Yazılım Testi Temeli.

Sınavınızın ardından BCS'den, tarafınıza sınavınız için kaydolmanız istendiği bir e-posta gönderilecektir. Kaydınızı olduktan ve sınav sonuçlarının da açıklanmasının ardından hesabınıza giriş yapabilir ve sonuçlarınızı görebilirsiniz.

Yöntem Tercihleri

BCS müfredatı için eğitim sağlayıcıların, bahsedilen genel yaklaşımlardan daha ayrıntılı bir şekilde açıklanması ve ele alınması gereken belirli bir geliştirme yaklaşımı seçmeleri gerekir. QA, tüm yaşam döngüsü boyunca geliştirme yaklaşımı olarak Dinamik Sistem Geliştirme Yöntemi'ni (DSDM) ve uygulama yazılımı geliştirme yaklaşımı olarak da Scrum'ı seçmiştir. Bu yaklaşımların her ikisi de sektörde popüler tercihler olup genelde birlikte kullanılırlar. Ayrıca modelleme işi ve yazılım spesifikasyonları için Yekpare Modelleme Dili (UML) de kullanılmaktadır.

Who Should Attend

Analiz konusunda yeni olanlar ve Sistem Geliştirmenin temel becerilerini edinmesi ve pratiğini yapması gerekenler.

Sistem Geliştirmenin Temel Unsurları alanında BCS Sertifikasına sahip olmak isteyenler. (Bu katılımcılar, SDEEX-2 kodlu eğitime de kayıt olmalıdırlar).

What You Will Learn

At the end of this course you will be able to:

- Identify the tasks and disciplines required for systems development and the implementation of the development
- Describe the relationship between systems development and the wider term solution development
- Interpret the business requirements and produce systems requirements
- Describe the commonly used development lifecycles defined in the syllabus
- Select a particular development lifecycle based on specific characteristics
- Describe in detail one method that embraces one (or more) of these generic lifecycles
- Describe the structure, activities and deliverables of this method
- Identify the key roles and responsibilities within the chosen method and describe how these can be used to form teams
- Describe, interpret and quality assure the key models that the selected method uses for defining the process, static and event processes of the system
- Explain the differences between logical and physical models
- Make effective use of different methods of interpersonal communications
- Quality assure the systems requirements documentation and models
- Identify different architectures for systems development solutions
- Conduct a quality review
- Explain how CASE, CAST and Application Management tools might be used to support the chosen method

Outline

Systems Development Roles and Responsibilities

Identify the Actors/Roles and responsibilities within system development and implementation (for example, analysts, designers, developers, testers and technical architects). Characteristics of these roles. Team structure.

Architecture

Different levels of architecture – Enterprise, Business, Solution, Infrastructure (networks, databases) Inputs at Enterprise level. Inputs at Solution and Infrastructure level. Impacts of design decisions.

Systems Development Lifecycles

Waterfall. V Model. Incremental or phased delivery. Spiral or iterative. Advantages and disadvantages of each approach. Selection of an appropriate approach on defined characteristics.

Development Practices

Bespoke development. Evolutionary / Agile. Prototyping. Component based development. Software Package solutions (COTS Commercial off-the-shelf).

Methods

Structure and content of a chosen representative method (DSDM/Scrum). Describe and interpret three representative models (UML) from the method, showing at least: Process perspective, Data perspective, Event perspective. Explain how these models can be used to depict the logical and physical aspects of a system. Roles and team structures within the chosen method. Products within a chosen method.

Systems Investigation

Fact finding approaches: Workshops, Prototyping, Interviewing, Questionnaires (for usability or package selection for example), Scenario Analysis. Functional requirements definition. Non-functional requirements definition. Documenting system requirements. Human aspects of systems investigation and introducing change.

System Design, Implementation and Maintenance

Controls and security. Verification and validation. Interface design. Design principles and constraints (legal, ethical, financial). Different types of implementation. Sign off and deployment. Post implementation reviews. Different types of maintenance and support.

Quality Assurance

Definitions of quality. Requirements driven testing. Types of walkthrough and inspection. Post Project Reviews. Service Level Agreements.

CASE, CAST and application lifecycle tools

Software support for systems development. Features of CASE and CAST tools. Features of Application Lifecycle Management. Lifecycle coverage. Configuration and version control.

Case Study and Exercises

Throughout the course a case study and other exercises are used to reinforce and practise the topics discussed. A sample exam paper with marking scheme is also provided.