

TM



EUCIP

European Certification of
Informatics Professionals

EUCIP

Systems Integration and Testing Engineer

Elective Level Profile Specification

Version 2.4, February 2007

Short Description

A EUCIP Systems Integration and Testing Engineer is expected to be very effective in several different areas of systems development: preparing end-user documentation, setting up IT systems, testing their functionality as a whole and as single component modules, identifying anomalies and diagnosing possible causes. Requirements include a specific knowledge on how interfaces between software modules are built.

This profile requires a minimum work experience of **36** months in a compatible job role; if this requirement is not fulfilled, the candidate might be certified as an **Associate** Systems Integration and Testing Engineer.

Tasks Overview

The Systems Integration and Testing Engineer works within organisations (either as an employee or as an external provider) to ensure that software systems and components are successful integrated across hardware systems and meet specified requirements.

Establishes correctness, completeness and security of IT systems assuring their quality and identifying potential improvement opportunities.

Participates in the definition of the organization's notion of software quality, according to the requirements expressed by managers, process owners and quality experts.

Selects and sets up a measurement framework characterizing software viability attributes, according to objectively quantified notions of software quality.

Produces detailed component specifications.

Analyses, recommends and selects as relevant, hardware/software platforms and products which ensure that applications conform to appropriate interconnection standards.

Recommends system integration application implementation strategies, selecting suitable component technologies and platforms.

Verifies integrated systems' capabilities and operation in relation to the organization's business goals.

Selects and develops integration testing techniques for ensuring the systems' overall capability of fulfilling the client's business requirements.

Participates in designing custom integration tests, reviewing functional specifications to identify open issues.

Selects and configures all necessary hardware and operating environments, at the appropriate level of automation and virtualization, as needed to efficiently complete all testing tasks.

Develops test plans and test procedures for white and black-box testing of all components at the unit, module, system and integration levels, performing risk analysis when required.

Establishes procedures for test results analysis and reporting in accordance with the organization process requirements. Resolves issues discovered during white-box testing.

Defines and handles defect tracking and correction procedures, monitoring and updating defect reports as needed

Writes test program to assess software quality, and develops test tools to increase test effectiveness.

Oversees product testing within a product unit, designs master test plans and schedules, and manages test organization.

Assists clients and users in defining both service levels and acceptance tests for automated systems. .

Monitors software systems post go-live to ensure they are operating as designed and takes full responsibility for quality assessment.

Identifies test design and implementation improvements, runs pilot projects to measure their impact, and drives their adoption across various development teams.

Reports on alternative test tools and environments and on the organisational and economic feasibility of various test solutions, advising the organisation on emerging technologies and practices in integration tests.

Produces high quality documents and written reports, describing test-related topics in a clear and concise style and in full accordance with the organization process.

Communicates with other ICT staff to monitor and assess the effectiveness of test execution and reporting, the efficiency of automated test solutions (in terms of ICT equipment, processing and communication capacity, etc.) and the technical feasibility of proposed new solutions.

Essential Behavioural Skills [2]¹

The Systems Integration and Testing Engineer role requires a good general knowledge, a strong will to learn, strong communication skills in writing.

Ability to fully analyze a problem and complete detailed reports.

Capability of working well in a team environment as well as alone, creating and managing a personal work plan to meet the project deadlines.

Ability to adapt to the changing needs of a project.

Initiative in gaining in-depth experience in areas of technology relevant to testing of company products.

Analytical and synthetic intelligence and proactivity are required to formulate and validate solutions.

A persistent focus on deadlines, flexibility, determination, planning and control aptitude are required to achieve results.

¹ numbers in brackets represent EUCIP points

Detailed Skills Required

Deep competence level [21,5]

B1.08 Software engineering principles [1,5]

- Understand roles of the software engineering process (project manager, software developer, maintenance staff, quality assurance and the user).
- Understand software development life cycle models and their applications.
- Understand and apply software development estimation techniques.
- Understand and apply principles of software Project Management.
- Understand Risk Management.
- Understand Quality Assurance.
- Understand Configuration Identification, Control and Auditing.
- Understand Configuration Status Accounting.
- Understand and apply Software Estimating Techniques and Metrics.

B3.05 Principles of testing [2,5]

- Explain the principles of Testing.
- Maintain the importance of Testing in the Lifecycle.
- Understand Dynamic Test Techniques.
- Apply Test Management Standards.
- Use Static Testing Techniques.
- Understand core testing terminology (e.g. Expected Results, Expected Information).
- Appreciate the economics of Testing.
- Perform High Level Test Planning.
- Organise User Acceptance Testing (UAT).
- Ensure Functional and Non-Functional UAT is completed.
- Contribute to Dynamic Testing (Black Box).
- Contribute to Test Management (e.g. organisation, estimating, resourcing).

B3.09 Testing management [1]

- Contribute to the organisation of penetration testing.
- Set up and debug complex test systems.
- Prepare and review integration regression test cases.
- Execute integration test cases.
- Report defects.
- Verify fixed defects.
- Select design elements for code review.
- Report elements which fail on code review criteria.
- Contribute to security code reviews.

B1.04 Tools and techniques for development, testing and implementation of IT systems [1,5]

- Use system development tools for business modelling, requirements management and acceptance testing.

- Use relevant tools for automated testing (e.g. CAST).
- Use tools to support systems implementation and testing (e.g. rollout tools).
- Use relevant tools to perform security code review.
- Take responsibility for "end user" training and support documentation.
- Collaborate with Service Management to establish a user support structure (e.g. IT Service Desk).

A5.01 Project Management essentials [2,5]

- Define the role of the various specialists in a typical project organisation structure (e.g. Rational Unified Process, PRINCE2, etc.).
- Contribute to the IS project plan for a given business scenario.
- Contribute to risk analysis of a project proposal, concentrating on business risk.
- Use standard approaches to evaluate a project plan from the business viewpoint.
- Assist in defining the phases within a project and the role of the business analyst in those phases.
- Assist in the creation of constraints and the definition of milestones, checkpoints and reviews for a project.
- Define Corporate Standards for the documentation of business analysis deliverables in a project.
- Contribute to quality assurance processes within a project, from a business perspective.

B1.01 System development lifecycles [2,5]

- Understand the differences between Business Analysis, Systems Analysis and Systems Design.
- Investigate and document an existing system.
- Produce a requirements definition for a business system.
- Create Business System Options and present them to the business.
- Identify tasks/disciplines involved in management of systems development.
- Justify the use of a specific systems methodology.
- Use relevant (to Business and Systems Analysis) development techniques.
- Explain the lifecycle of a project to business users.
- Use formal approaches for ensuring best practice in the System Development process.
- Understand the rationale for a particular Systems Development (SD) method and where it is used.
- Appreciate the scope and limitations of SD method in the project lifecycle.
- Understand and work within a standard development framework (e.g. SSADM).
- Appreciate the need for specific techniques in the SD process.
- Evaluate the suitability of differing system development approaches for a particular project scenario.

- Harmonise roles and responsibilities of the various specialists in each of the main lifecycles for system development.
- Use well known approaches to providing detailed SD Lifecycle products, e.g. textual, diagrams, prototypes.
- Create different modelling views of a business system (e.g. static data, behaviour, user centred, process).

B1.05 Systems design and implementation [1,5]

- Identify the tasks involved in implementing and designing an IT system.
- Evaluate the business benefits of database technologies, data warehousing and data mining tools.
- Understand the contents of a system specification.
- Understand function specifications.
- Appreciate the need for (and constraints on) Physical Design of Databases (e.g. tables and indexes).
- Perform Forms Design for a business system.
- Contribute to design of screens and dialogues.
- Contribute to recovery and contingency plans.
- Ensure that audit of an Information System is possible.
- Define system controls for an Information System.
- Define the data integrity needs for an IT System.
- Understand Technical System Options and assist the business in evaluation.
- Employ relevant methods of changeover to new systems.
- Contribute to System Review (post implementation).
- Detail the need for design of security, confidentiality and privacy in a system.
- Produce an implementation plan and assist with business implementation and system review.

B1.09 Computer Aided Software Engineering (CASE) and Integrated Development Environment (IDE) tools [1]

- Know when and how to use a CASE tool: top issues related to CASE tools adoption, CASE Tools for different platforms/languages.
- Work with the most used IDEs for Windows and Unix platforms.
- Integrate plug-ins in an IDE. Examples: Oracle Developer 2000, Rational ROSE, Select, Business Objects.
- Customise the build process in an IDE.
- Use the "Configuration Manager" (Debug/Build...).
- Integrate the IDE with a Version Control System. Example: CVS.

B1.10 "Dry run" application testing [1,5]

- Explain the principles of Testing.
- Apply Test Management Standards.
- Understand core testing terminology (e.g. Expected Results, Expected Information).
- Perform High Level Test Planning.
- Organise User Acceptance Testing (UAT).

- Prepare reasonable and meaningful data sets that allow a realistic simulation of the future live system.
- Specify a comprehensive set of procedures and different application flows that shall be tested.
- Perform Dynamic Testing (Black Box).
- Test system functions with real sample and simulated data sets.
- Conduct functional and non-functional UAT, and ensure full user acceptance of both the system and the proposed data models.
- Record possible emerging issues and classify them by type and severity.

B1.11 Enterprise applications integration [2]

- Explain the characteristics and typical features of enterprise application software in comparison with personal productivity software, system software and web services.
- Identify the actual and potential scope of each of the software applications that compose the Information System (IS) of the target organisation.
- Identify functional overlaps between applications, describe related inefficiencies and risks and propose effective solutions.
- Envision sensible uses of web interfaces and automated e-mail responders for a wider and secure access to information stored in enterprise-level applications (like ERP or legacy systems).
- Provide guidance on how a standard package can be localised to fit languages, conventions and specific statutory requirements or other typical local requirements.
- Use internal system tools, macro procedures, query languages or package-specific customising environments (e.g. ABAP/Netweaver for SAP) to adapt the software functions to specific needs.
- Write a functional specification of software interfaces that can contribute to the optimal integration of the IS.
- Distinguish between temporary and permanent interfaces, one way or “to-and-from” data flows, real-time or periodic alignments, incremental or regenerative synchronisations.
- Distinguish between database-level data load (e.g. through SQL instructions or import functions), stored procedures and application-level bulk load (e.g. through redirection of input).
- Perform testing and basic troubleshooting in the usage of interfaces between software systems.

B1.14 System deployment methods [1,5]

- Organize the deployment of a system, i.e. the delivery of it to the users in the target client organisation.
- Control and understand the business application of artefacts resulting from software development.
- Organise deployment workflow and product roll-out activities, including:
 - o testing the software in its final operational environment (beta test),
 - o packaging the software for delivery,
 - o software distribution,
 - o software installation and configuration,
 - o data population, both through new data entry activities and through

- migration from legacy system files or databases,
 - training the users.
- Support the client organisation in planning and acting the operational start-up of the new system.
- Organise and control initial support service provision during system start-up.

B3.08 Writing technical documentation and procedures [1,5]

- Understand documentation writing in multicultural and multilingual environments.
- Observe and make the most effective use of corporate standards for documents.
- Prepare templates for shared documents.
- Define naming conventions and classification criteria.
- Write and proof read software documentation.
- Write processes and procedures descriptions.
- Compose and publish information manuals.
- Develop software product and project presentations.
- Publish documentation on a web.

B3.07 Build reports [1]

- Administer server resources.
- Create high quality web reports.
- Use templates to create mailing labels and letters.
- Create and modify basic tabular reports.
- Build reports using XML.
- Add dynamic data to an HTML page.
- Identify the main components in a report document and how they are related.
- Publish a report on the web.
- Tune reports.
- Create other report styles such as break reports and matrix reports.
- Use report parameters and customise a runtime parameter form.
- Manage report templates.
- Create and embed a graph in a report.
- Identify standard report design styles and run existing reports to various output destinations.

Incisive competence level [8,5]

B3.04 Designing and developing distributed and critical applications [1]

- Choose the right level of transaction support.
- Plan and design for performance, maintainability, extensibility, availability, scalability, and reliability. Considerations include:
 - number of transactions per time increment,
 - bandwidth,
 - capacity,
 - peak versus average usage requirements,

- response-time expectations,
- barriers to performance,
- processes per server,
- parallel execution,
- maintenance expectations,
- location and knowledge level of maintenance staff,
- impact of third-party maintenance agreements,
- hours of operation,
- level of availability,
- impact of downtime,
- growth of the partners,
- growth of the company,
- volume of documents.
- Design integration with existing applications. Derive the physical design.
- Install remote components: considerations include attended and unattended installations.
- Troubleshoot failed installations.
- Identify situations for applying custom components.
- Monitor and optimise performance: tools include performance counters, Event Viewer, Windows Management Instrumentation (WMI).
- Diagnose and resolve implementation errors.

B2.01 Information modelling techniques and tools [1]

- Investigate existing systems and define elements of logical data design for required systems.
- Contribute to schema definition for a given business scenario.
- Use both top down and bottom up modelling of data.
- Use entity relationship modelling or class modelling to outline the information requirements of a new business system.
- Use recognised entity modelling techniques to construct a data model reflecting the business needs of the organisation.
- Contribute to data key identification and design.
- Assist in the creation of a Data Catalogue.
- Understand the concepts of RDA.
- Validate data models from a business processing perspective.
- Specify the requirements for security and integrity of data in a business scenario; requirements should cover integrity, loss of data consistency, logical errors, system errors, hardware failures, human error.

A1.02 Requirements engineering [1]

- Distinguish between Functional and Non Functional requirements.
- Use What, Why, How questioning to elicit requirements.
- Differentiate between requirements and project constraints.
- Identify the Actors in the Requirements Management process: Domain Expert, End User, Requirements Engineer, and Developer.
- Perform requirements elicitation.
- Perform Problem and Business understanding activities.
- Understand the needs and constraints of stakeholders.

- Use Creative thinking and related techniques (e.g. interviews and scenarios, observation, prototyping, workshops, generic requirements for industry sector).
- Prioritise Requirements (e.g. 80/20, MoSCoW, Needs and Musts).
- Resolve overlapping requirements.
- Judge whether a problem is a cause or symptom.
- Resolve conflicting requirements.
- Reduce ambiguity of requirements .
- Ensure Testability of requirements.
- Support requirements validation via reviews and prototyping.
- Achieve Requirement Refinement.
- Manage the requirements definition process.
- Differentiate between stable and volatile requirements.
- Apply versioning principles to requirements documents.
- Establish traceability and ownership of requirements.
- Use CASE Tools for requirements management.
- Act as an effective member of a team involved in eliciting and recording user requirements for an Information System.
- Apply a range of elicitation techniques effectively.

A4.03 Package implementation techniques [1,5]

- Acquire an understanding of the software package market in a particular business context.
- Evaluate a software package against defined requirements.
- Document the functional match of a package solution.
- Perform a gap analysis for a package selection.
- Present the recommendations for a specific package solution.
- Present recommendations concerning the "fit" of the software package to agreed functional and non-functional requirements.
- Explain the advantages of the package approach.
- Identify, assess and propose solutions for enhancing end-user productivity through:
 - o optimal use of package software,
 - o integration between different package software modules (if applicable),
 - o integration between package software and common desktop applications,
 - o personalized user interface and restricted data sets.
- Appreciate the issues with tailoring the package software.
- Identify specific requirements to be solved through custom software development and write functional specifications for necessary changes or extensions to the standard package.
- Explain a specific package implementation methodology (e.g. ASAP for SAP).
- Explain the different scope of subsequent modelling and piloting phases of a typical package implementation methodology.
- Define the modified business processes required in a package solution.
- Ensure that business events are managed through appropriate systems and that adequate information is distributed to the decision makers according to the projected business processes.

- Ensure that the target organisation is aware of all security issues related to a complex information system.
- Configure package security features so as to grant the right data visibility to every user and to prevent unauthorized access.

A4.02 Package selection and Implementation Lifecycle [1]

- Define a framework for effective package selection.
- Identify, investigate and assess potential package suppliers.
- Evaluate a software package against defined requirements.
- Present recommendations concerning the "fit" of the software package to agreed functional and non functional requirements.
- Evaluate the advantages and disadvantages of the package approach.
- Evaluate the human, technical and financial implications of a decision to outsource development/buy a package solution.
- Apply a checklist of factors to a decision on in house development vs. package procurement.
- Work within a framework for package selection.
- Understand the impact on package selection of Prototyping approaches.
- Acquire an understanding of the software package market in a particular business context.
- Produce a High Level Functional Model for a system.
- Contribute to identifying potential package suppliers.
- Contribute to the production of Invitations to Tender (ITTs) and questionnaires.
- Investigate suppliers.
- Assist in the creation of Supply Contracts and Support Agreements.
- Perform cost comparisons - purchase and support.
- Document the functional match of a package solution.
- Contribute to gap analysis for a package selection.
- Use a weighted scorecard approach to evaluation.
- Present the recommendation for a specific package solution.
- Assist in the implementation of packages.
- Liaise with procurement staff for package purchase.
- Define the modified business processes required in a package solution.
- Appreciate the issues with tailoring the package software.
- Contribute to long term supplier management.
- Appreciate the advantages/disadvantages of packages.

B1.02 User centred analysis and development [1]

- Analyse and contribute to the design of Information Systems which reflect the way users wish to work to support their business.
- Perform user analysis and establish usability criteria, which can be used to measure the success of new systems.
- Model business tasks and use these models as a basis for prototyping and user interface design.
- Use User Analysis, Work Practice Models, Task Modelling, Job Design, or equivalent techniques.

- Appreciate the role of User Centred techniques in System Development.
- Perform Work Practice Modelling using concepts such as actor, task, business event, task scenario, user role, user class, user object, common subtask.
- Map Business Activities onto the organisational structure.
- Create required task models, plans and scenarios.
- Contribute to identifying and specifying task-supporting IT software functions.

A5.02 Estimating for system development [1]

- Use a variety of estimating approaches and apply them to a practical project.
- Understand the importance of estimating and measurement.
- Distinguish between top-down and bottom-up estimating.
- Contribute to “estimating by analogy”.
- Contribute to Delphi estimating.
- Contribute to estimating by the analysis percentage effort method.
- Appreciate the principles of Function Point Analysis (FPA).
- Contribute to FPA estimates by using formal counting rules.
- Assist in defining effort estimates and elapsed duration estimates.
- Appreciate the use of Line Count Cost Models.
- Contribute to building Work Breakdown structures and hence estimating for software development projects.
- Appreciate the impact of timeboxing and RAD on estimating.
- Evaluate the factors affecting productivity in IS development.
- Contribute to collecting and analysing project statistics/metrics.
- Contribute to the use of metrics to improve project estimation.

B1.12 Defining a solution architecture [1]

- Gather and analyse:
 - o user requirements,
 - o operational requirements,
 - o system requirements for hardware, software, and network infrastructure.
- Transform requirements into functional specifications: considerations include performance, maintainability, extensibility, scalability, availability, deployability, security, and accessibility.
- Transform functional specifications into technical specifications: considerations include performance, maintainability, extensibility, scalability, availability, deployability, security, and accessibility.
- Select the appropriate technologies for the physical design of the solution.
- Create the physical design for:
 - o the solution,
 - o deployment,
 - o maintenance,
 - o the data model.
- Create specifications for auditing and logging.
- Validate the physical design.

External references to SFIA[®] version 3 by the SFIA Foundation

Skill 31: Systems Integration

“The incremental and logical integration and testing of components and/or subsystems and their interfaces in order to create operational systems.”

Levels 4 and 5

Skill 25: Systems Testing

“The planning, design, management, execution and reporting of tests, using appropriate testing tools and techniques and conforming to agreed standards, to ensure that new and amended systems, together with any interfaces, perform as specified.”

Levels 4 and 5

Skill 22: Programming / Software Development

“The design, creation, testing and documenting of new and amended programs from supplied specifications in accordance with agreed standards.”

Levels 4 and 5

Skill 32: Porting / Software Integration

“The integration of software products into existing software environments to produce new platform-specific versions of the software products.”

Levels 4 and 5

Skill 37: Business Process Testing

“The planning, design, management, execution and reporting of business process tests and usability evaluations. The application of evaluation skills to the assessment of the ergonomics, usability and fitness for purpose of defined processes. This includes the synthesis of test tasks to be performed (from statement of user needs and user interface specification), the design of an evaluation programme, the selection of user samples, the analysis of performance and inputting results to the development team.”

Levels 4 and 5

External references to AITTS by the German Government – *Arbeitsprozessorientierten Weiterbildung in der IT-Branche*

Profil 2.4: IT Test Coordinator (IT-Testkoordinator/in)

“IT Test Coodinator konzipieren die Test, die den Software- und Hardware-Entwicklungsprozess begleiten, auf den Stufen Unit-, Integrations-, Funktions-, System- und Akzeptanztest und führen diese Tests durch. Sie stellen Testumgebungen bereit und sind für die Tests auf allen Teststufen verantwortlich.”

Profil 2.5: IT Technical Writer (Dokumentationsentwickler/in)

“IT Technical Writer erstellen und pflegen Dokumente von IT-Projekten und von IT-Produkten im Lebenszyklus.”

External references to *Nomenclature 2005* by CIGREF (club informatique des grandes entreprises françaises)

Métier 4.3: Intégrateur d'applications

“Sous la responsabilité du chef de projet maîtrise d'oeuvre, il participe au choix des différents composants logiciel (progiciels, bases de données, développements spécifiques...) et en assure l'assemblage dans le respect du plan d'urbanisme des systèmes d'information de l'entreprise et de l'architecture retenue pour le projet. En ce qui concerne les développeur, soit en externe avec l'aide d'une société de services.”