

Process to design an IoC curriculum

Introduction

This document outlines a process to develop a degree curriculum that will satisfy the IoC standard. The worked example is based on Bachelor's degree for a "web developer" profile, which is entirely arbitrary, but should be sufficiently widely understood to serve as a generic example.

The IoC standard

The IoC standard for a Bachelor's degree with honours specifies that students should:

- have demonstrated the responsibility characteristics for SFIA Level 3;
- have demonstrated competence in one or more relevant SFIA skills at level 3 (the competence skill(s));
- have underpinning knowledge for a total of four SFIA skills at levels 3 or 4, including the competence skill(s), and with at least one at SFIA level 4;
- understand and apply the legal, social, ethical, and professional issues, and drivers for sustainable computing that are relevant to the chosen skills;

The demonstration of competence requires that students have not done things just once or twice, in a controlled/safe environment, but have had experience sufficient to be competent to deploy the skills for a real customer, without needing routine supervision.

In order to achieve competence in any SFIA skill, graduates must have demonstrated the responsibility characteristics for SFIA Level 3. Demonstrating these characteristics requires evidence of experience from completing a number of independent projects within the scope of the chosen SFIA skill together with evidence of generic leadership and business skills.

Since the underpinning knowledge required for at least one skill at SFIA Level 4 may be for the same SFIA skill as that for the "competence" skill, the requirement for underpinning knowledge in this case would be for three additional skills at SFIA Level 3. If the underpinning knowledge at SFIA Level 4 is for a SFIA skill different from the "competence" skill, then the requirement for underpinning knowledge is for one skill at SFIA Level 4 and two further skills at Level 3, in addition to that for the "competence" skill.

The computing profession is subject to a range of legal and ethical requirements which may be specific to particular skill areas – such as accessibility law – or may be more generic – such as the provisions of GDPR. Graduates should be fully aware of the legal and ethical requirements that will affect them, as well as the social and professional responsibilities that arise in their work. It is likely that sufficient detail of legal and ethical requirements will have been taught alongside technical material for graduates to have a sound understanding of how these requirements affect their activities. In addition to these explicit requirements, it is likely that any graduate achieving the standard will also have underpinning knowledge, and possibly competence, for several SFIA skills at Levels 1 or 2.

Outline process

1. Identify SFIA skills that would be relevant for the required role
2. Review the selected skills against industry / academic requirements – address any gaps
3. Review the skills identified to determine which should be output (SFIA Level 3) requirements and which are needed as "underpinning" at SFIA/HE levels 1/2
4. Refine the list of output SFIA skills until there are 4 or 5 at SFIA level 3. This may involve either "demoting" skills if too many are deemed relevant or adding additional generic skills should there be too few.
5. For each of the selected SFIA skills, identify what knowledge is required to underpin the skill
6. Identify what prior knowledge would be required for each package of knowledge

7. Identify opportunities for experience development that could be incorporated into the curriculum (to allow students to build evidence of competence).
8. Package the learning into modules, to create a curriculum
9. Check learning dependencies between modules
10. Map contributions from each module to output SFIA skills.

Process – worked example

In this section, we work through the development of the "web developer" degree programme summarised in the [Appendix: Web Developer Example](#).

1. Identify SFIA skills

Identify SFIA skills that would be relevant for the required role. The initial list of SFIA skills is constructed simply by considering each of the skill areas in turn.

- Information security
- information content publishing
- data management
- portfolio, programme and project support
- requirements definition and management
- software design
- programming/software development
- data modelling and design
- Testing
- information content authoring
- user experience design
- user experience evaluation
- systems integration and build
- service design - service level management

- service transition - configuration management, asset management, change management, release and deployment
- service operation - system software, security administration, application support, IT infrastructure, database administration, storage management, network support, problem management, incident management
- Quality management
- Quality assurance
- Measurement
- stakeholder management - supplier management, customer service support
- sales and marketing - marketing, sales support, product management

This list is clearly too long, and some of the skills identified are areas for which only superficial knowledge may be needed.

2. Review the selected skills

Review the selected skills against industry / academic requirements against industry / academic requirements.

Reviewing the skills identified, one further skill was added – Solution Architecture. This is defined only from level 4 in SFIA v7, but something in this area is important underpinning for any web developer.

3. Review the skills identified

Review the skills identified to determine which should be output (SFIA Level 3) requirements and which are needed as "underpinning" at SFIA/HE levels 1/2.

The third stage is to refine the list of possible skills under four headings:

- Level 3/4 target
- Level 2 underpinning
- Only awareness needed
- Not relevant

Level 3/4 target

- Information security SCTY
- requirements definition and management REQM
- programming/software development PROG
- systems integration and build SINT
- release and deployment RELM

Level 2 underpinning

- Solution Architecture ARCH (not currently defined at level 2)
- portfolio, programme and project support PROF
- data modelling and design DTAN
- information content authoring INCA
- user experience design HCEV

Awareness only

- service design - {service level management}
- service transition - {configuration management, asset management, change management, release and deployment}
- service operation {system software, security administration, application support, IT infrastructure, database administration, storage management, network support, problem management, incident management}

Not relevant

Note that some of the skills identified under this heading are "not relevant" simply because they are subsumed in other skills already selected.

For example, user experience evaluation is part of user experience design.

- information content publishing
- data management
- software design

- Testing
- Quality management
- Quality assurance
- Measurement
- user experience evaluation
- stakeholder management - supplier management, customer service support
- sales and marketing - marketing, sales support, product management

4. Refine the list

Refine the list of output SFIA skills until there are 4 or 5 at SFIA level 3. This may involve either "demoting" skills if too many are deemed relevant or adding additional generic skills should there be too few.

In this iteration, there are 5 target skills for levels 3 or 4. Hence, the set of target skills should both be achievable and offer sufficient breadth.

5. Identify underpinning knowledge

For each of the selected SFIA skills, identify what knowledge is required to underpin the skill. For underpinning knowledge, look at the skill description for Level 3. In some cases, the description of the activities are already fairly close to a description of the knowledge required, for example, Information content authoring Level 2 INCA2. For others, the underpinning knowledge may simply be, "know how to...".

For example, for [Data Modelling and Design DTAN L2](#):

Applies data analysis, design, and modelling techniques to establish, modify or maintain a data structure and its associated components (entity descriptions, relationship descriptions, attribute definitions).

The relevant underpinning knowledge might be:

Know how to apply data analysis, design, and modelling techniques to establish, modify or maintain a data structure and its associated components.

This will include understanding how to construct entity descriptions, relationship descriptions and attribute definitions.

6. Identify prior knowledge

Identify what prior knowledge would be required for each package of knowledge. Some skills may need to be contextualised.

For example, Programming/Software Development PROG is generic, but this particular programme is for web developers.

So, **PROG L3**:

Designs, codes, verifies, tests, documents, amends and refactors moderately complex programs/scripts. Applies agreed standards and tools, to achieve a well-engineered result. Collaborates in reviews of work with others as appropriate.

Needs to be interpreted in the context of web development – both server-side and client-side scripts, in a range of languages, and probably using databases and other technologies. Thus, the underpinning knowledge is likely to be built on introductory programming, display elements, scripting languages, database definition and manipulation and an understanding of the architecture options for web apps. This is standard curriculum construction – high level outputs need to be grounded in lower level skills.

7. Identify experience opportunities

Identify opportunities for experience development that could be incorporated into the curriculum (to allow students to build evidence of competence). Building competence through experience – and accumulating evidence of that experience – is central to an IoC degree.

There are many ways in which opportunities for real experience may be built into a degree programme. The mechanisms chosen for this worked example are:

- Two "clinic" modules, in which students, under supervision, provide website construction/development for (paying) local customers

- Industry-sponsored and mentored group projects in years 2 and 3
- Summer internships between years 1 & 2 and/or years 2 and 3
- A final year project that should deliver at least one product for a "real" customer.

8. Package into modules

Package the learning into modules, to create a curriculum. The draft curriculum is shown [here](#)

With the exception of the higher than usual proportion of experience modules (necessary to deliver competence) and the fairly tight focus on web development, the curriculum actually looks fairly traditional. It may lack a little breadth compared with a traditional programme, and happens, in this example, to lack any explicit options. However, there will be considerable choice for students in how they focus their activities in the experience modules.

9. Check learning dependencies

This is, again, standard curriculum design. The dependencies between modules are shown in the "builds on" / "supports" links in the module descriptions.

10. Map to SFIA skills

Map contributions from each module to output SFIA skills.

The purpose of the model curriculum is to ensure that students meet the IoC standard. A good final check in the design of the curriculum is to ensure that the knowledge required for each of the target skills is covered, probably across several modules, and that there are ample opportunities for appropriate experience to be gained. It is helpful here to consider the links from the skill descriptions to the modules that underpin them.