

# Re-architecting Software Development



Dr Donna O'Shea, Lecturer in Computer Science

Architecture  
regardless of the  
domain is about  
structure & vision.

# Software Architecture

## Application Architecture

Building blocks are software based.

Concerned about programming languages, constructs, libraries, frameworks.

## System Architecture

Multiple applications across multiple tiers and technologies.

End to end system at a high level.

Concerned with issues such as interoperability and integration with other systems.

## Software Architecture

Practices that help us build better software.

Object oriented principles, inversion of control, refactoring, automated unit testing, clean code.

## Enterprise Architecture

Breadth rather than depth, strategy rather than code.

How best to use technology across an organisation without getting into detail about how that technology works.

# Software Architecture versus Design?



**Shape of system**  
**Structure of system**  
**Technology Choice**  
**Framework Choice**  
**Design pattern choice**

Does every software  
project need software  
architecture?

# **Software Architect**

It is a role, not a rank.  
It forms part evolutionary process  
where a software developer  
gradually gains the experience and  
confidence to undertake the role.

# Software Architect

- Technical leadership & communicator.
- Understanding of software development process.
- Understanding of business domain.
- Technology knowledge.
- Design & Programming skills.



**“I want to talk to somebody who actually writes software, not a box-drawing hand waver.”**

# MSc Software Architecture & Design

# MSc in Software Architecture & Design

Software Architecture & Design (M)	Software Vulnerabilities (E)
Software Process Engineering (M)	Decision Analytics (E)
Scalable Microservices (M)	Research Methods (M)
Data Analytics (E)	Research Project (M)
Programming Language Design (E)	Declarative & Concurrent Programming (E)

# System Architecture

# Software Architecture

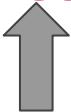
# Application Architecture

Scalable Microservices (M)	Software Architecture & Design (M)	Software Process Engineering (M)	Algorithm Design & Analysis (M)
In this module the student will learn a framework technology and using this framework will develop skills to build, automate, scale and manage a distributed system created from a number of collaborating components represented as microservices.	In this module, students will learn about the principles of software architecture and design and how and when to apply software design and architecture patterns to solve common problems when designing and developing software with the goal of creating an extensible and maintainable software solution.	In this module, students will be introduced to the main software development methodologies and DevOps processes designed to support the development of software and software based systems. As part of this module, students will examine the challenges that architects, developers and testers face in this environment and strategies and best practices to minimise risk and maximise the quality of the software produced.	This module explores techniques for the analysis and design of efficient algorithms. In this module the learner will be introduced to a set of graph algorithms available for solving network flow problems, shortest paths, minimum spanning trees, and graph connectivity. It also covers the strengths and limitations of approximation algorithms to tackle hard computational problems (NP-hard).

# System Architecture

# Software Architecture

# Application Architecture



Declarative & Concurrent Programming (E)	Programming Language Design (E)	Software Vulnerabilities (E)	Decision Analytics (E)
<p>In this module, learners will be introduced to parallel, concurrent and distributed programming, which are key concepts for correctly designing distributed-based applications and architectures. As part of this module students will be introduced to an industrial declarative programming language called Erlang, which is a programming language that has built in support for concurrency, distribution and fault tolerance.</p>	<p>In this module, the student will design and implement a language processor using different architectural and design patterns. As part of the module the student will complete a practical software development project that combines areas in software engineering such as formal languages, grammars, automata, software architecture and design, programming, operating systems, computer structure and model-driven development.</p>	<p>In this module students will learn how to design and develop software so that security is an integral part of the process. In addition, students will learn how to locate and identify software vulnerabilities and how to prevent the inclusion of such vulnerabilities in software.</p>	<p>In many real-life applications with underlying constraints there are several parameters to configure. Solving such problems involves finding configurations that satisfy the constraints associated to the problem. In addition, many times there are certain criteria to optimise. In this module, students will learn to model optimisation problems and apply algorithms that facilitate decision making leading to optimal solutions.</p>

Email:

**Donna.OShea@cit.ie**

Website:

**cs.cit.ie**

This Presentation:

**<https://goo.gl/5Q306j>**

