POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG8430E: Software Architecture and Advanced Design

Fall 2020

Instructor

- Marios Fokaefs – Assistant Professor
- Research
  - DevOps
  - Software Evolution and Maintenance
  - Cloud computing, IoT, SOA
  - Software Engineering Economics
- https://www.polymtl.ca/expertises/en/fokaefs-marios-eleftherios
- marios.fokaefs@polymtl.ca
- Office M-4119
  - Virtual Webex: https://polymtl.webex.com/meet/marios.fokaefs

Instructor – Marios Fokaefs

- BSc in Computer Science – Greece
- MSc in Software Engineering – Alberta
  - Identification of Extract Class refactoring opportunities in object-oriented systems.
- PhD in Software Engineering – Alberta
  - Web service evolution
  - Economic management of web services
- Research Assistant (postdoc) – York
  - Self-adaptive cloud systems
  - Economic management of cloud and distributed systems.

Mohammadreza Rasolroveicy - TA for Software Architecture

- PhD Student at Polytechnique
- Research interests: DevOps - AutoScaling - Blockchain - IoT Security - Cloud Computing
- Email : mohammadreza.rasolroveicy@polymtl.ca
- Office : 4218 - Department of Computer & Software Engineering , Polytechnique Montréal

Course Objectives

- Present the styles and patterns of software architecture and design putting an emphasis on distributed systems.
- Measure and maintain the quality of design and architecture.
- Explore and master the design of distributed systems in real case studies: frameworks, cloud and service-oriented systems, and systems for big data analytics.

At the end of the course, the student will be able to:

- Design the architecture of software by choosing the styles of architecture and design patterns according to the requirements and justifying these decisions.
- Design the components of the architecture by using special models and novel technologies (event-driven architectures, service-oriented and object-oriented architectures, cloud)
- Evaluate the quality of an architecture or a software design and maintain it by applying changes.

Situation of the course

- LOG8430 ends a chain of courses:
  - INF1005C – procedural programming
  - INF1010 – object-oriented programming
  - LOG2410 – software design
  - LOG3210 – Elements of languages and compilers
  - LOG3430 – Methods of software testing and validation

Situation of the course

- LOG8430 is also a graduate course
- Be at ease to read and write in English.
- Conduct research
  - There is not always one single correct answer.
  - Justify and verify your responses.
  - Critical thinking
  - Collaborate!
- Work well in a team.
- Advanced concepts on:
  - Object-oriented design
  - SOA
  - Cloud
  - Distributed Systems

Format of the course

- 2 hours of lecture – 1 hour of in-class practice
  - Exercises on the course of the day.
  - In teams
  - Bring your own laptops or portable devices
- RECOMMENDATIONS!
  - Option 1: We start at 9am until 11:20 without breaks
  - Option 2: We start at 8:35 until 11:20 with two 10-minute breaks.
- 3 hours of lab every two weeks
  - Work on the assignments.

Format of the course

- 2 hours of lecture [illegible] of in-cla[illegible]ctice
  - Exercises on the cou[illegible]day
  - In teams
  - Bring your own laptops or p[illegible]evices
- RECOMMENDATIONS!
  - Option 1: We start at [illegible]til 11:2[illegible]t breaks
  - Option 2: We start [illegible] until 11:20 w[illegible]0-minute breaks.
- 3 hours of lab eve[illegible]wo weeks
  - Work on the assignments.

Virtual format of the course

- Pre-recorded lectures on YouTube.
  - You are invited to watch the videos before the actual day of the course and prepare your questions for discussion.
- A Slack space will be maintained for direct interaction.
  - There will be channels for lectures, labs, assignments and exams.
  - You can post your questions any time, but the space will be closely monitored during course hours.
- Live interactive sessions will be held over Webex for specific lectures that contain “in-class” exercises.
  - These exercises are based on the course of the week and they highly represent questions that will be in the quizzes or the final exam.
  - Live sessions will be two hours long (from 10am to noon) and if there is time left after the exercises we will get a chance to discuss questions about the course.

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Course schedule

Course	Content	Date	Live Webex	Deadline
01	Introduction Basic concepts SOLID principles Design patterns	3 September	Yes	
02	Distributed architectures Architectural styles and patterns Frameworks Ecosystems	10 September	Yes	
03	Design quality Quality attributes Quality metrics	17 September	Yes	
04	Bad design Antipatterns / design smells Refactoring, refactoring to patterns	24 September	No	

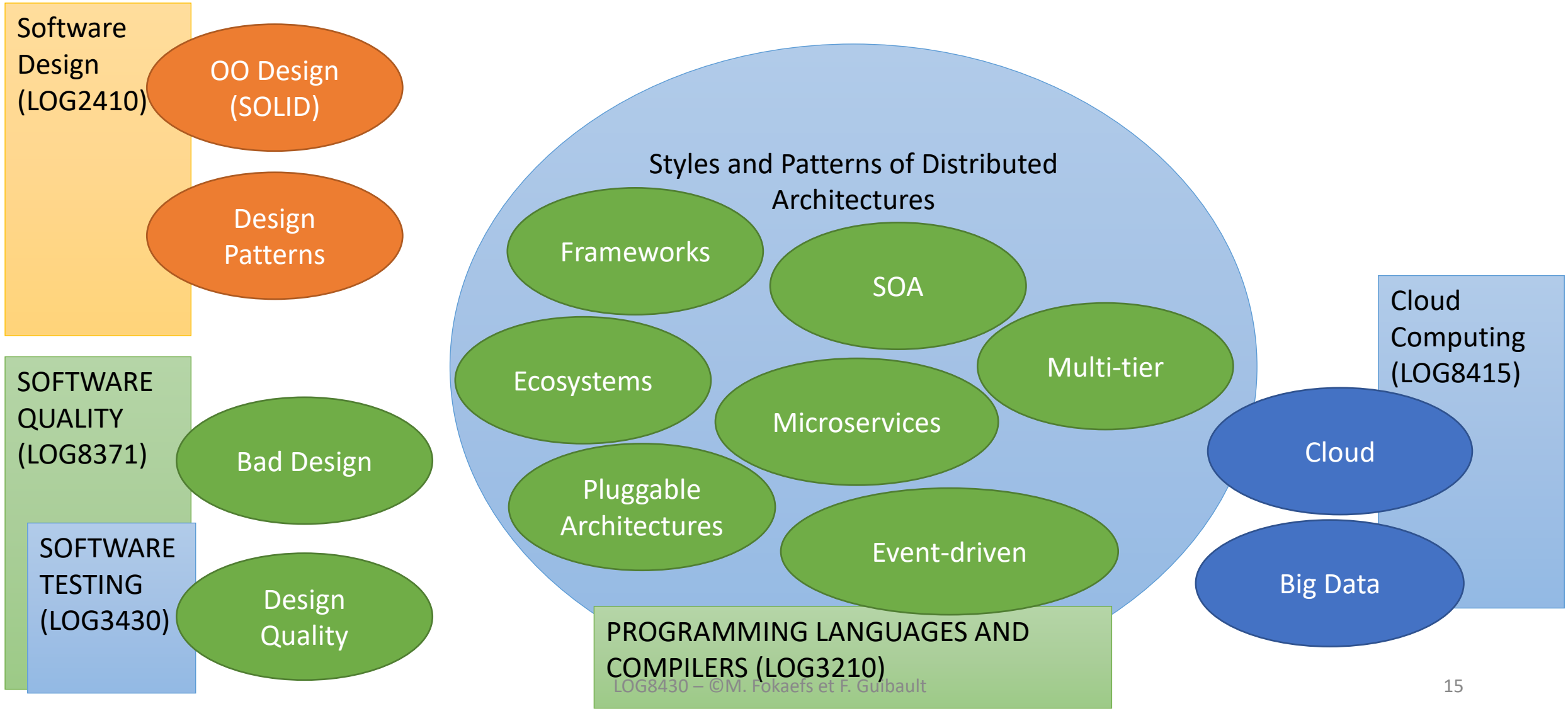
Course schedule

Course	Content	Date	Live Webex	Deadline
05	Software Architecture for big data: Insertion and analytics	1 October	Yes	TP1
06	Software Architecture for big data: Databases	8 October	Yes	
--	*Reading Week*	15 October	No	
07	Blockchain	22 October	Yes (?)	
08	Event-driven architectures Concurrency patterns	29 October	Yes	TP2

Course schedule

Course	Content	Date	Live Webex	Deadline
09	SOA	5 November	Yes	
10	Microservices	12 November	Yes (?)	
11	AOP	19 November	No	
12	Assignment presentations	26 November	Yes	TP3 (presentations)
13	Revision	3 December	Yes (?)	TP3 (papers)

Course Map


Software Design (LOG2410)
OO Design (SOLID)
Design Patterns
SOFTWARE QUALITY (LOG8371)
Bad Design
SOFTWARE TESTING (LOG3430)
Design Quality
Styles and Patterns of Distributed Architectures
Frameworks
SOA
Ecosystems
Multi-tier
Microservices
Pluggable Architectures
Event-driven
PROGRAMMING LANGUAGES AND COMPILERS (LOG3210)
Cloud Computing (LOG8415)
Cloud
Big Data

Resources

- There is no principal textbook, but many resources to study and understand the material:
  - Links
  - Tutorials
  - Textbooks
  - Articles
  - Software
- Available on Moodle for every class.

Tools

- Eclipse - https://www.eclipse.org/
- Git
  - GitHub - https://github.com/
  - Bitbucket - https://bitbucket.org/
- Design and modeling
  - Papyrus - https://www.eclipse.org/papyrus/
  - ArgoUML - http://argouml.tigris.org/
  - Enterprise Architect - https://sparxsystems.com/products/ea/
  - Understand - https://scitools.com/features/
- Source code analysis
  - JDeodorant - https://marketplace.eclipse.org/content/jdeodorant
  - Ptidej - http://www.ptidej.net/
  - SonarCloud - https://sonarcloud.io/

Important

- All the material presented during the lectures and the labs can be the object of an evaluation.
- The content of the course is dynamic!
- All slides are currently available in Moodle.
  - Changes may occur in the slides.
- If there are updates on the material, there will be announcements on the forum.
- The videos will be uploaded as the course progresses.
- Late submission of assignments will receive 10% penalty per day of delay.
- Any request to differ an exam needs to pass through the office of academic affairs.
- Collaboration is permitted for the assignments, but all rules and regulations relevant to plagiarism apply at all time.

Evaluation of the course

- Assignments 60%
  - TP1 15%
  - TP2 15%
  - Discussion assignment 30%
- Quizzes 3 18% (6% each)
- Final exams 22%
- At least 40% in the final exams to pass the course.

Assignments

- In teams of 3 people
  - Use the link for “Group Formation for Assignments” in Moodle by September 10.
- Objectives
  - Master the theory
  - Understand and analyse
  - Design and decisions
- 2 practical assignments + 1 discussion assignment (paper)
- 2 lab sessions per assignment: 1st will show tutorials, 2nd will answer questions and solve problems.
- Practical assignment: Analysis of a system (architecture recovery, problems in design and so on).
  - Prepare a report of the analysis.
- Discussion assignment: Compare some solution alternatives, exercise some systems to evaluate them.
  - Prepare a paper describing a core concept, the comparison of alternatives and the experimental evaluation.

Final exams and Quizzes

- Individual
- 3 Quizzes throughout the term
  - Possible dates: Sept 17, Oct 1, Oct 22
  - 20 points each (6% of the total course grade)
- Final exams: 3 questions
  - Each question is worth 20 points (22% of the total course grade)
  - Total 60, maximum 40.
- The total points between quizzes and final exams are 120 points (100 max + 20 points bonus).
  - This means you can either miss one quiz or you can answer only two of the 3 questions of the final exam.
  - If you miss one quiz and you answer all 3 questions of the final exam, the extra points will be the grade for the quiz you missed. E.g., You missed quiz 2 and you got 54 in the finals, the grade for your final will be 40 (perfect score) and for your 2 quiz, 14/20.
  - Extra points in the final cannot be used to improve grade in quizzes.
- Theory and practice
  - Like the exercises in class.
- Examples of systems discussed in class, in lab or in the assignments.
- Quizzes and exams will be performed as Moodle Tests online.

Interactive sessions

- Exercises and quizzes on the subject of the week.
  - Use of breakout rooms and brainstorming discussions
- Learning by practice.
- Improve and evaluate your ability to make, explain, and justify decisions on problems around software design and architecture.
- There will be no evaluation for these exercises.
- BUT! Examples of such exercises may be used in the final exam.

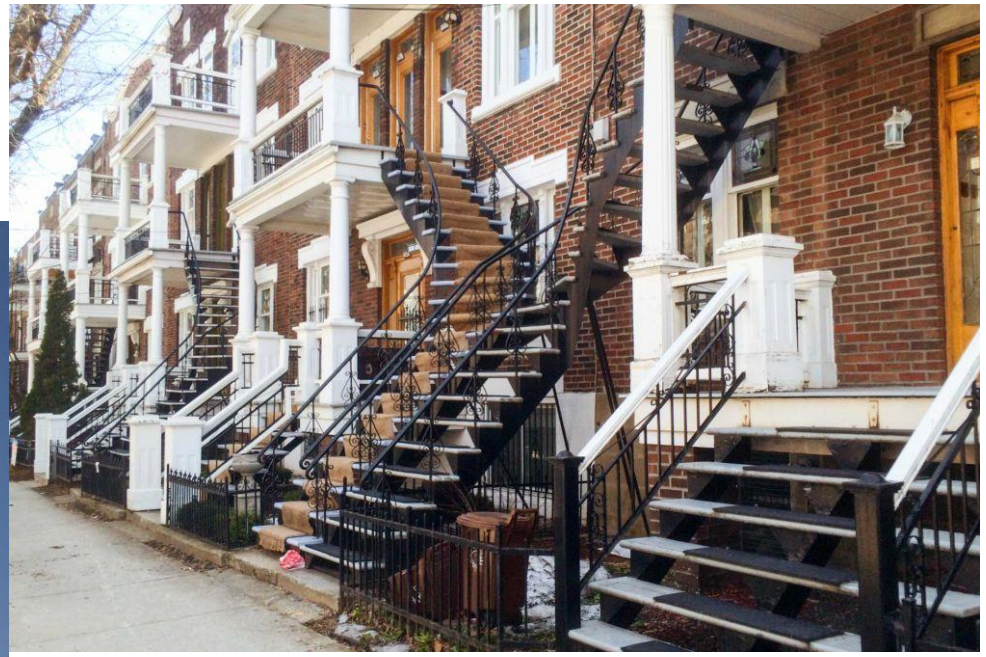
3 Questions on software architecture

1. What is a software architecture?
2. What are the criteria that guide a software architect to choose a specific architectural style?
3. What impact does the choice of architecture have on the development of a software system?

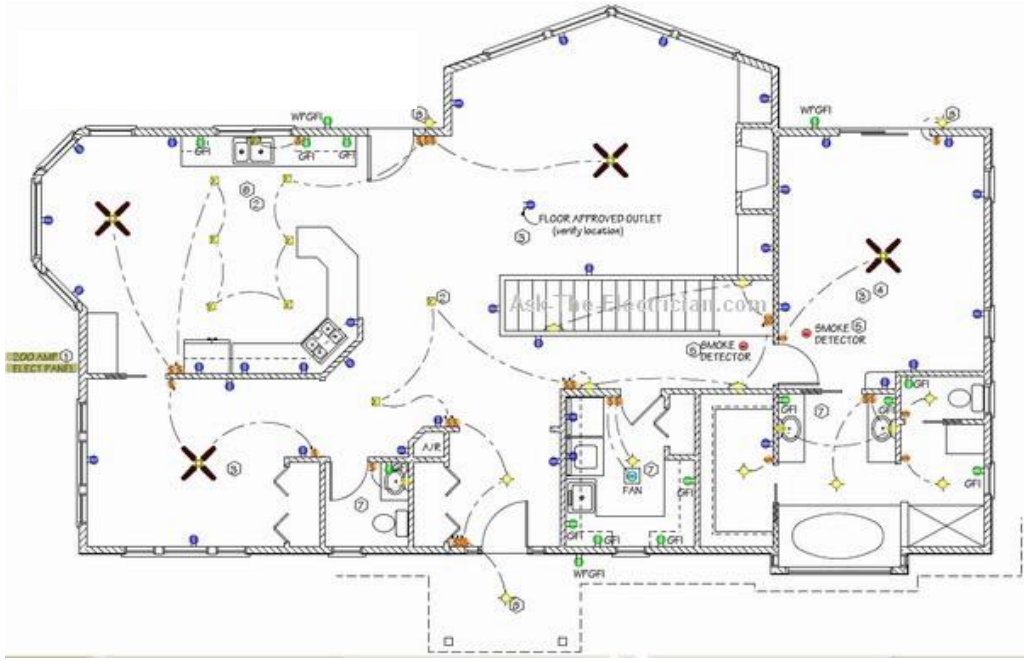
3 Questions on software design

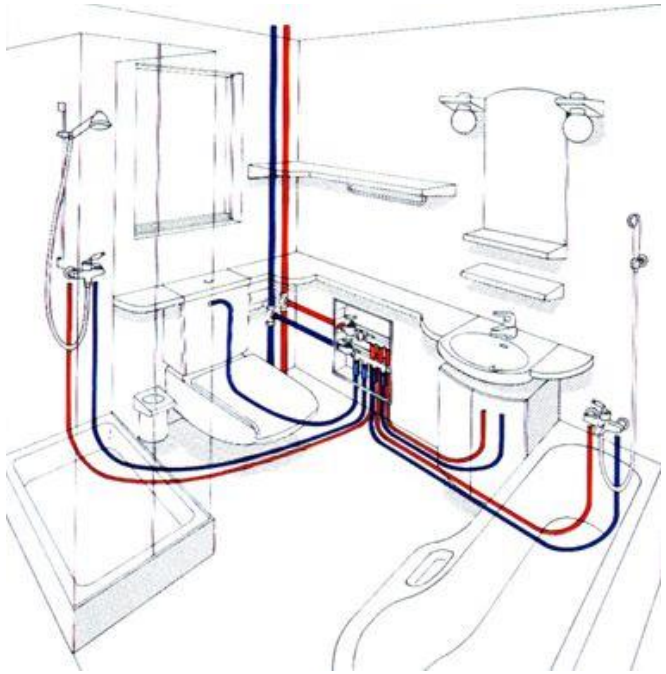
1. What is the difference between software architecture and software design?
2. Are the decisions on architecture and design independent?
3. How can we evaluate the quality of an architecture and the quality of a design?

Architecture and design in the building domain



Architecture and design in the building domain





3 Questions on software architecture

1. What is a software architecture?
2. What are the criteria that guide a software architect to choose a specific architectural style?
3. What impact does the choice of architecture have on the development of a software system?

3 Questions on software design

1. What is the difference between software architecture and software design?
2. Are the decisions on architecture and design independent?
3. How can we evaluate the quality of an architecture and the quality of a design?

What is architecture?
What is design?

Software Architecture (ANSI/IEEE Std 1471-2000)

The architecture of a software system is defined as the fundamental organization of the system, embodied in its components, their relationships with each other and their environment, and the principles that guide its design and evolution.

Software Architecture (SEI definition)

The architecture of a software system is the set of structures necessary to reason about the system, which include the software elements, the relations between them and the properties of each one of them.

Software Architecture (Sangwan, 2015)

The architecture of a software system is fundamentally concerned with the organization of a system into its constituents and their interrelations in order to achieve a given objective.

Architecture

- It answers to “what?”
- On a model level
  - Packages, components and relationships
- A more abstract level of decisions
  - The decisions that need to be made in advance.
  - The decisions which are very expensive to change during development.
- Relevant to the operational objectives and the non-functional requirements.
- It includes:
  - Modules
  - Databases and data technologies
  - Execution and deployment environments

Design

- It answers to “how?”
  - How are we going to implement the architecture?
- More concrete elements:
  - Classes
  - Methods/Functions
  - Data tables
- Relevant to the functional requirements.
- Elements that can change during development (maintenance phase)
  - Reengineering (refactoring)
  - Evolution
  - Adaptation

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Architecture or Design?

Next time

Software Design (LOG2410)

OO Design (SOLID)