

---

# Software Architecture and Design

Werner Dietl <wdietl@uwaterloo.ca>

## Table of Contents

1. Overview .....	1
2. Contact .....	1
2.1. Teaching assistants .....	2
3. Course expectations .....	2
4. Overview of topics .....	2
5. Course material .....	2
6. Course schedule .....	3
7. Project .....	4
8. Assessment .....	5
9. Policies .....	5
10. Acknowledgments .....	7

## 1. Overview

*Software Architecture and Design (SE464)*<sup>1</sup> "Introduces students to the design, implementation, and evolution phases of software development. Software design processes, methods, and notation. Implementation of designs. Evolution of designs and implementations. Management of design activities." Course catalog.<sup>2</sup>

## 2. Contact

Lectures are held **Tuesday and Thursday from 10:00 to 11:20 in MC 4061**. Tutorials are held **Friday from 13:30 to 14:20 in EV3 1408**. There are no scheduled lab slots. Students are expected to organize project meetings on their own.

My office hours are **by appointment** and will be held in my office, EIT 4007. I will also be available immediately after lectures. See contact details<sup>3</sup>.

Course material, announcements, and submissions will be handled through Learn<sup>4</sup>.

Begin all email subjects with [SE464].

Try not to leave your questions until the last minute.

---

<sup>1</sup> <http://www.adm.uwaterloo.ca/cgi-bin/cgiwrap/infocour/salook.pl?level=ugrad&sess=1179&subject=SE&cournum=464>

<sup>2</sup> <http://www.ucalendar.uwaterloo.ca/1718/COURSE/course-SE.html#SE464>

<sup>3</sup> <https://ece.uwaterloo.ca/~wdietl/contact.html>

<sup>4</sup> <https://learn.uwaterloo.ca/d2l/home/337191>

## 2.1. Teaching assistants

The course is assisted by:

- Erlie Shang [eshang@](mailto:eshang@)
- Parsa Pourali [ppourali@](mailto:ppourali@)

Meetings by appointment.

## 3. Course expectations

It is expected that students attend lectures and tutorials and complete the required assignments. Lectures will often include hands-on activities; participation in these exercises is essential to succeed in the class. Slides will be provided via Learn. Any material discussed in class (lectures & tutorials) or in the required readings will be testable unless otherwise noted.

By the end of the course you should be able to:

- propose and analyze software architectures.
- explain the strengths and weaknesses of various architectural styles and design patterns / techniques.
- communicate and rationalize architectural and design decisions.
- ideate, justify, and implement software designs.
- evaluate, compare, and contrast different architectures and designs.

## 4. Overview of topics

- Software architecture, architectural styles, and architectural representations
- Software design, design patterns, design representations
- Software architecture and design conception, analysis, and communication
- Architecture and design recovery / reverse engineering
- Architecture and design visualization / understanding
- Cloud / grid computing architectures

## 5. Course material

While the course does not have a required textbook, much of the materials will be sourced from the first two texts; additional books are supplementary.

- Richard N. Taylor, Nenad Medvidovic, and Eric Dashofy. *Software Architecture. Foundations, Theory, and Practice*. Available in the library or for purchase (e.g., through Amazon.ca<sup>5</sup>). Slides for this book are available online<sup>6</sup>.
- Ian Gorton. *Essential Software Architecture*. Available online<sup>7</sup> or for purchase (e.g., through Amazon.ca<sup>8</sup>). Slides for this book are available online<sup>9</sup>.
- Fred P. Brooks Jr. *The Mythical Man Month*. Available in the library or for purchase (e.g., through Amazon.ca<sup>10</sup>).
- Fred P. Brooks Jr. *The Design of Design*. Unfortunately not in the library but still available e.g. through Amazon.ca<sup>11</sup>.
- *The Architecture of Open Source Applications*. Available online<sup>12</sup>.

## 6. Course schedule

Lecture material is available through Learn<sup>13</sup>.

This is a tentative schedule that might get adapted during the term.

Week	Class
1	Introduction, organization, and project
2	Decomposition, Non-functional properties, Architecture
3	Architectural Styles
4	Architectural Styles
5	Modeling, Design Patterns
6	Design Patterns (no lecture on Oct. 11)
7	Frameworks
8	Architecture of Open-Source Applications
9	Dependency Injection, Cloud/REST Architectures

<sup>5</sup> [http://www.amazon.ca/gp/product/0470167742?](http://www.amazon.ca/gp/product/0470167742?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0470167742)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0470167742](http://www.amazon.ca/gp/product/0470167742?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0470167742)

<sup>6</sup> <http://www.softwarearchitecturebook.com/resources/>

<sup>7</sup> <http://bit.ly/9RF18P>

<sup>8</sup> [http://www.amazon.ca/gp/product/3540287132?](http://www.amazon.ca/gp/product/3540287132?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=3540287132)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=3540287132](http://www.amazon.ca/gp/product/3540287132?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=3540287132)

<sup>9</sup> <https://docs.google.com/viewer?url=http://www.ug.it.usyd.edu.au/~iango/home/ESA-Slides.pdf>

<sup>10</sup> [http://www.amazon.ca/gp/product/0201835959?](http://www.amazon.ca/gp/product/0201835959?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201835959)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201835959](http://www.amazon.ca/gp/product/0201835959?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201835959)

<sup>11</sup> [http://www.amazon.ca/gp/product/0201362988?](http://www.amazon.ca/gp/product/0201362988?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201362988)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201362988](http://www.amazon.ca/gp/product/0201362988?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201362988)

<sup>12</sup> <http://aosabook.org/en/index.html>

<sup>13</sup> <https://learn.uwaterloo.ca/d2l/home/337191>

Week	Class
10	Service-oriented-Architectures, Microservices
11	Languages: Scala, Checker Framework, TypeScript, etc.
12	Project presentations
13	Project presentations, outlook, and wrap-up

## 7. Project

The project forms an integral part of this course. The goal of the project is to produce a significant application that performs some useful function (even better, something **awesome!**). This software must have a considered and defensible design and architecture.

There is only one real restriction on the application idea itself: apps that require crowd buy-in are not acceptable (e.g., apps that would require large numbers of people to contribute content to be viably useful). Simple database management apps (e.g., simple CRUD apps) are strongly discouraged.

Mobile applications have been popular in previous years. To make the architecture interesting, aim to make the app executable on at least two platforms, e.g. a mobile phone app and a browser interface. Using sensors and external devices could also make for interesting apps.

However, other project ideas are welcome: if you have an idea for a project of a suitable size and complexity and find a project team to work with, propose it!

The projects will be completed in teams of four. You are free to select your own team; if you do not have a team or your team has less than four members, try using the course forums to organize a team. If all else fails, please talk to me and I will set you up. Each of the deliverables for the project can be considered an assignment.

Projects will have a difficulty scale applied to them by the instructor and TAs. The scale formula will be:

$$(\text{project} + \text{bonus}) * \text{scale} = \text{final project grade}$$

Scale will range between 0.75 and 1.0. The components of the scaling mark will be determined by:

- 5 completeness (compared to proposal)
- 5 utility
- 5 polish
- 10 difficulty

There will also be various sources of bonus marks during the term; each will be worth 2%:

- Best proposal
- Best prototype demo

- Best final demo
- Accepted to curated App Store/store/public website

Note: The expectation is that you will work approximately 12 hours per week on this course; at least 8 of these hours will be on the project. Given that the course lasts 13 weeks, each team member is expected to work on the project at least 100 hours. You should be able to accomplish something pretty great in this time; please make the most of this opportunity.

## 8. Assessment

Deliverable	Date	Format	Value
Design background	Sept 11	Learn	Pass/Fail
Project groups	Sept 15	Learn	Pass/Fail
D1: Proposal	Sept 22	Learn + TA	5%
D2: Prototype	Oct 16	Learn + TA	5%
Midterm	Oct 24	In class	10%
D3: Draft Arch + Design	Nov 6	Learn + TA	Pass/Fail
D4: Presentation	Nov 21 — 30	In class	10%
D5: Arch + Design	Dec 1	Learn + TA	20%
D6: Participation journal	Dec 1	Learn	Pass/Fail
Final Exam	TBD	Written	50%

You must pass the final exam and all pass/fail assignments to pass the course.

No late submissions will be accepted.

It is expected that all team members receive the same grade for the project component. The participation journal will be used to arbitrate disagreements.

## 9. Policies

### Academic Integrity

- In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. [See the academic integrity<sup>14</sup> site for more information.]

---

<sup>14</sup> <http://www.uwaterloo.ca/academicintegrity/>

- Turnitin.com: Text matching software (Turnitin®) will be used to screen assignments in this course. This is being done to verify that use of all materials and sources in assignments is documented. Students will be given an option if they do not want to have their assignment screened by Turnitin®. In the first week of the term, details will be provided about arrangements and alternatives for the use of Turnitin® in this course.

## Grievance

- A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70<sup>15</sup>, Student Petitions and Grievances, Section 4.
- When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

## Discipline

- A student is expected to know what constitutes academic integrity<sup>16</sup> to avoid committing an academic offence, and to take responsibility for his/her actions.
- A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean.
- For information on categories of offences and types of penalties, students should refer to Policy 71<sup>17</sup>, Student Discipline.
- For typical penalties check Guidelines for the Assessment of Penalties<sup>18</sup>.

## Appeals

- A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground.
- A student who believes he/she has a ground for an appeal should refer to Policy 72<sup>19</sup>, Student Appeals.

## Note for Students with Disabilities

- AccessAbility Services<sup>20</sup>, located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to

---

<sup>15</sup> <http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm>

<sup>16</sup> <http://www.uwaterloo.ca/academicintegrity/>

<sup>17</sup> <http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm>

<sup>18</sup> <http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm>

<sup>19</sup> <http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm>

<sup>20</sup> <https://uwaterloo.ca/disability-services/>

lessen the impact of your disability, please register with the AccessAbility Services at the beginning of each academic term.

## 10. Acknowledgments

Thanks to Derek Rayside and Reid Holmes for sharing their experience and materials from previous iterations of this and a similar course, CS 446, Winter 2014<sup>21</sup>.

---

<sup>21</sup> <https://cs.uwaterloo.ca/~rtholmes/teaching/2014winter/cs446/index.html>