

17-633 and 17-882 Architectures for Software Systems

[M/W/F 9:30am-10:50am]

[Spring 2024, 12 Units with 6 Unit version in A3]

Instructor	Email	Office Location and Hours
Prof. David Garlan	garlan@cs.cmu.edu	Wednesdays, 4-5 Remote)
Dr. Bradley Schmerl	schmerl@cs.cmu.edu	Wednesdays, 11-12 TCS 421)
Swathi Rai	swathir@andrew.cmu.edu	Fridays, 11-12 TBD (SCR 264)

Course Description. *Architectures for Software Systems* teaches you how to design, understand, and evaluate systems at an architectural level of abstraction. By the end of the course you will:

- Understand the technical, organizational, and business role of software architecture in software engineering.
- Recognize how architectural drivers influence the choice of architectural designs, and will be able to identify and define the key drivers for realistic software systems.
- Identify key architectural structures (styles, patterns, tactics, etc.).
- Generate architectural alternatives in a given context and choose among them. Apply principles of good architectural documentation and presentation to describe a software architecture and the rationale behind its design.
- Understand the impact that open source and third party components have on architectural designs, including ways to integrate mismatched elements.
- Understand how formal notations can be used to specify architectures.
- Evaluate the fitness of an architectural design in meeting a set of system requirements and balancing quality tradeoffs.
- Be aware of the future trends in software architecture.

Prior Knowledge. Students should have had experience developing medium- to large-scale software, preferably in an industrial and team-based setting.

Lectures and Recitation: There will be two lectures and one recitation weekly. Attendance at lectures and recitations is mandatory.

Computing: A personal computer or laptop is required for this course. For some course assignments we will be using the Java programming language. You will need to download and install the Java Software Development Kit (the latest version of J2SE). You may use any editor or development environment that you like.

Course Meetings: All lectures and recitations will in person.

Class: Monday/Wednesday, 9:30AM - 10:50PM

Recitation: Fridays, 9:30- 10:50PM

Learning Resources.

Required Text: *Software Architecture in Practice, Fourth Edition*, by Bass, Clements, Kazman, Addison-Wesley 2021 [[BCK21](#)].

We will also use a collection of supplementary readings, which will be available through the course web site.

Reference/Auxiliary Texts: (not required, but worth owning)

- *Architecting Software Intensive Systems: A Practitioner's Guide*, Anthony Lattanze, Auerbach 2008 [Lat08].
- *Documenting Software Architectures: Views and Beyond, Second Edition*, by Paul Clements, et al. Addison-Wesley 2011 [C+11].
- *Software Architecture: Perspectives on an Emerging Discipline*, Mary Shaw and David Garlan, Prentice-Hall, 1996. [SG96]

Assessments. Student learning is enhanced through applying and explaining ideas to others; thus, the course includes the following activities:

Assessment	Description	Grade %
Quizzes	There will be a short quiz at the start of each recitation on the material of the week. No late or make-up quizzes. The score of the lowest quiz will be dropped.	15%
Assignments	There will be 4 assignments. 2 assignments will be team assignments. Each will be allocated 1 week to complete. Late assignments will be penalized 10% per day.	40%
Project	Each team will produce a final project focusing on a preliminary architectural design for a relevant system, and an in-depth analysis of several architectural design choices for it.	40%
Instructor discretion	The instructors reserve a portion of the grade to account for positive and negative aspects of student performance not otherwise measured above. This includes things like general level of effort, and peer reviews from the assignments and project, as well as attendance at lectures and recitations and level of engagement such as asking and answering questions. The quality of contribution is more important than the quantity.	5%

Note that regardless of whether students are taking the 6 unit legacy version of the course or the 12 unit version, the distribution of grades will be the same.

Course and Grading Policies

- **Submission policy:** assignments and projects should follow the following:
- Submitted assignments should be named `LastName-FirstName-AssignmentNumber` (where `LastName` = your last name, `FirstName` = your first name, and `AssignmentNumber` = the assignment number), or for team projects, `TeamID-AssignmentNumber` (where `TeamID` = the team id and `AssignmentNumber` = the assignment number). Example: `Team1-Assignment1.docx`
- The time that the assignment is uploaded will be counted as the submission time.
- Assignments are to be submitted via Canvas.
- Assignments should be in either Microsoft Word or pdf format
- **Late-work policy:** Work is expected to be handed in at the respective due date and time. Late assignments will be penalized 10% per day.
- **Participation policy.** Class participation will be graded by in-class engagement, including asking relevant questions based on a critical review of required readings, lectures, and comments made by your peers.

Accommodations for Student Disabilities. If you have a disability and have an accommodations form from the Disability Resources office, we encourage you to discuss your accommodations and needs with us as early in the semester as possible. We will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, we encourage you to contact them at access@andrew.cmu.edu.

Academic Integrity. Honesty and transparency are important to good scholarship. Plagiarism and cheating, however, are serious academic offenses with serious consequences. If you are discovered engaging in either behavior in this course, you will earn a failing grade on the assignment in question, and further disciplinary action may be taken. For a clear description of what counts as plagiarism, cheating, and/or the use of unauthorized sources, please see the [University's Policy on Academic Integrity](#). If you have any questions regarding plagiarism or cheating, please ask us as soon as possible to avoid any misunderstandings. For more information about Carnegie Mellon's standards with respect to academic integrity, you can also check out the [Office of Community Standards & Integrity](#) website.

Use of Generative and other AI tools:

The point of being a software architect is to have a deep understanding of architecture drivers, design alternatives, and trade-offs to make educated judgements about the design of software. We recognize that generative AI tools are now a fact of life, and can be helpful if used appropriately. But they can not substitute for the creative and critical thinking you need to be an architect. You are permitted to use them in assignments in class, subject to some restrictions:

- **Attribution:** If you use AI tools to help you in your project or personal reflection paper, you must include a brief statement (e.g., as comments in the code or as an appendix to the paper) saying how they were used.
- **Responsibility:** You must not assume that anything you get from an AI tool is correct or appropriate; you must verify the correctness and appropriateness before using it.
- **Understanding:** You must not use AI as a substitute for learning. If you get something from an AI tool, you must understand it before you use it, and we reserve the right to ask you to explain your work.

If you're not sure whether some use of AI tools is acceptable, ***seek advice from the instructors.***

You are not permitted to use them for the open book quizzes.

Student Wellness. As a student, you may experience a range of challenges that can interfere with learning, such as strained relationships, increased anxiety, substance use, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may diminish your academic performance and/or reduce your ability to participate in daily activities. CMU services are available, and treatment does work. You can learn more about confidential mental health services available on campus at the [Counseling and Psychological Services](#) website. Support is always available (24/7) from Counseling and Psychological Services: 412-268-2922.

Respect for Diversity. We intend that students from all backgrounds and perspectives be well served by this course, that students' learning needs be addressed both in and out of class, and that the diversity that students bring to this class be viewed as a resource, strength, and benefit. We intend to present materials and activities that are respectful of diversity: gender, sexuality, disability, age, socioeconomic status, ethnicity, race, and culture. We encourage and appreciate your suggestions. Please let us know if any of our class meetings conflict with your religious observations so that we can make alternate arrangements for you.

Course Schedule. Note that bold face indicates readings in the course textbook:
[\[BCK21\]](#)

#	Date	Major Topic	Lecture	Assignments	Readings
1	W Jan 17	Architecture Overview	What is software architecture		
2	M Jan 22		Basic Architecture Concepts	A1 Assigned	Ch 1-2 [GS94] pp.1-5
3	W Jan 24	Architecture Requirements	Architecture Drivers and Quality Attributes		Ch 3
4	M Jan 29	Styles, Platforms, Product lines, Ecosystems	Dataflows and Events	A1 Due, A2 Assigned	[GS94] pp.5-16
5	W Jan 31		Call/Return and Repository		[Mic09], Ch3Li
6	M Feb 5		Platforms, Frameworks, Product Lines, and Ecosystems	A2 Due, A3 Assigned	Ch 17 [Bos09]
7	W Feb 7	Architecture Design	Tactics		Ch 4, 8, 9
8	M Feb 12	Techniques and Practices	Architecture Evaluation		Ch 21, [Mar+05][Nyg11] [KR18]
9	W Feb 14		Principles of Architecture Documentation		Ch 22, [Bro22]
10	M Feb 19		Architecture Decision Records, Michael Keeling, Kiavi	A3 Due, Project Assigned	[GS06], [B+18 skim]
11	W Feb 21	Architecture in Practice and Future Directions	Modeling and Analysis		Ch 24.3
12	M Feb 26		Architectures for Intelligent Systems, Grace Lewis, SEI.		[VOX21]

13	W Feb 28		Architecture in the 21 st Century		[Bro22.]
	F Mar 1		Mid-Semester Project Due. End of 6 unit legacy course.		
SPRING BREAK					
14	M Mar 11	Advanced Topics	Tactics 2	A4 Assigned	
15	W Mar 13		Architecture modeling in C4, AADL		
16	M Mar 18	Case Studies	Architecture Perspectives on DevOps, Len Bass	A5 Assigned	
17	W Mar 20		ROS Style Case Study		
	F Mar 22		In-Class A4 Presentations	A4 Due	
18	M Mar 25	Architecture and Code	Architecture Hoisting		
19	W Mar 27		Frameworks 2 and Mismatch		
20	M Apr 1		Architecture Recovery	A5 Due, Final Project Assigned	
21	W Apr 3		Architectures at Run Time		
22	M Apr 8		Formal Modeling and Architecture		
23	W Apr 10		Final Project Progress		

24	M Apr 15		Specifying Safety for Autonomous Driving.		
25	W Apr 17		TBD		
26	M Apr 22		PhD Presentations		
27	W Apr 24		Future Trends and Research		

Recitations and Quizzes

#	Date	Recitation	Quiz
1	F Jan 21	Thinking like an Architect	Quiz 1 - Architecture Introduction
2	F Jan 26	Architecture Drivers	Quiz 2: Concepts and Drivers
3	F Feb 2	More Architecture Drivers	Quiz 3 - Architecture Styles
4	F Feb 9	Tactic Practice	Quiz 4 - Platforms and Architecture Tactics
5	F Feb 16	Architecture Documentation (Homework)	Quiz 5 - Evaluation and Documentation
6	F Feb 23	First Project Progress Review	Quiz 6 - Practices
7	F Mar 15	Architecture and AI Exercise	Quiz 7 - Tactics/Modeling
8	F Mar 22	A4 Presentations	Quiz 8 - Case Studies
9	F Mar 29	Architecture and Code	Quiz 9 - Architecture and Code
10	F Apr 5	Recovery and Run-time Use	Quiz 9 - Recovery and Run time
11	W Apr 10	Final Project Progress Review	Quiz 10 - Formal models
	F Apr 12	NO RECITATION - SPRING CARNIVAL	
	F Apr 19	NO RECITATION - INSTRUCTORS AT CONFERENCE	
12	F Apr 26	Final Project Presentations	

References

GS94	Introduction to Software Architecture David Garlan and Mary Shaw. CMU Technical Report CMU-CS-94-166. 1994.
Bos09	From Software Product Lines to Software Ecosystems Jan Bosch.
Bro22	The lost art of software design. Simon Brown. Devovx 19th Edition, Belgium, 2022.
Fai22	Two Kinds of Iteration Links to an external site. . G. Fairbanks, IEEE Software 39(1), 2022
Fai21	Why Is It Getting Harder To Apply Software Architecture? . G. Fairbanks, IEEE Software 38(4), 2021.
Mar+05	Architecture Reviews: Practice and Experience . J. Maranzano, S. Rozsypal, G. Warnken, D. Weiss, P. Wirth, and G. Zimmerman. <i>IEEE Software</i> 2005
GCSS04	Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure . Garlan, Cheng, Schmerl, Steenkiste. <i>IEEE Computer</i> 0018-9162/04/2004.
Gar14	Software Architecture: A Travelogue . David Garlan, ACM978-1-4503-2865-4/14/05.
GS06	Architecture-driven Modelling and Analysis . David Garlan and Bradley Schmerl, 2006.
KR18	Share the Load: Distribute Design Authority with Architecture Decision Records . Links to an external site. Michael Keeling and John Rundle. Agile18, 2018.
Mic09	Microsoft Application Guide, Chapter 3: Architecture Patterns and Styles . Links to an external site. Microsoft, October 2009.
Nyg11	Documenting Architecture Decisions , by Michael Nygard. 2011.
VB19	Vogelsang, Andreas, and Markus Borg. " VB19.pdf Download VB19.pdf ." In Proc. of the 6th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2019.