



Carnegie Mellon University  
Master of  
Software Engineering

### 17-635: Software Architecture

Lectures: TR 7:00 - 8:20pm ET, Remote

Recitation: T 12:00 - 12:50pm ET, Remote

Course Zoom Link:

<https://cmu.zoom.us/j/95322536306?pwd=S3Q0Ym5melJwaEVZQjdmNTU3N3hMUT09>

Meeting ID: 953 2253 6306

Passcode: 834097

[A3, Spring 2022, 6 Units, Section 43]

**Last Update: 12/23/21**

**Instructor**  
Paulo Merson

**Email**  
[pmerson@cmu.edu](mailto:pmerson@cmu.edu)

**Office Location & Hours**  
Zoom; by appointment

### Teaching Assistant

*There's no TA for this section*

**Course Description.** The general goal of the Software Architecture course is to give students a broad yet sound view of what is software architecture, how it is created, documented, and used in practice.

This course draws fundamental concepts from a vast body of theoretical knowledge available about software architecture, and complements that with key information and best-practices to be successful working with software architecture in industrial projects at any scale.

This is a lecture-based course, and students will work on several exercises with practical focus. Class size will be limited.

Specific objectives of this class include:

- Exercise the creation and evaluation of software designs from three different perspectives: implementation units; runtime components; deployment.
- Recognize the importance of quality attributes and learn tactics and patterns that help to realize quality attribute requirements of performance, availability, usability, modifiability, and others.
- Learn how to document an architecture using multiple views, using informal notations and UML.

In support of these objectives students will be exposed to several example designs, with an opportunity to discuss the tradeoffs of key design decisions. Students will be expected to demonstrate understanding of theoretical concepts as well as the ability to make sound design decisions and communicate architectural design they create to others.

**Prior Knowledge.** Students are expected to be familiar with programming (in any programming language). Prior experience in the development of medium- or large-scale software systems is desirable, but not required.

### Learning Resources

**Required textbooks** (ebooks can be purchased at [www.informit.com](http://www.informit.com), discount code: BUY2):

- L. Bass, P. Clements, R. Kazman. *Software Architecture in Practice, Third Edition* (or Fourth Edition). Addison-Wesley, 2013.
- P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond, Second Edition*. Addison-Wesley, 2010.

Lecture slides and other course materials will be distributed online and accessible with a CMU account via Canvas. Classes will be recorded via Zoom so that students in this course (and only students in the course) can watch or rewatch past class sessions. Recordings will be made available on Canvas as soon as possible after each class session, and will be accessible on Canvas. You are not allowed to share these recordings. This restriction is to protect your FERPA rights and those of your fellow students.

We will be using Piazza for asynchronous class discussion. Rather than emailing questions to the instructor, we encourage you to post your questions on Piazza

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

**Assessments.** Students learn more by applying and explaining ideas to others, thus, the course requires the following activities:

- **Weekly quizzes:** These are short online questions derived from the reading assignments and lectures, to be answered individually.
- **Presentation:** Students will give a short presentation about a text (blog post, article, paper, book chapter, etc.) that is relevant to the class.
- **Class participation:** Students are stimulated to contribute questions, answers and comments during class, and participate in online discussions in Piazza.
- **Final exam:** open book timed examination.

Assessment	Final Grade % (yes, the total is 102%)
Six weekly quizzes	48%
Presentation	10%
Class participation	8%
Final exam	36%

---

### Course and Grading Policies

- **Late-work policy:** All work is expected to be handed in at the indicated due date and time. For fairness to the whole class, no late submissions or makeups will be accepted for the quizzes.
- **Participation policy.** Class participation will be evaluated by in-class engagement, including asking relevant questions based on a critical review of required readings and lectures, and responses on the class online discussion board. The lack of attendance, punctuality, and participation will count against your participation grade.

## Course Schedule

The following schedule provides a general overview of topics and assignments, but it is subject to change.

Week	Topics	Assignments	Notes
1	<ul style="list-style-type: none"><li>● Course introduction</li><li>● History of Software Architecture</li><li>● Architecture: Motivation and definition</li><li>● Architecture and Agile</li><li>● Architecture drives</li></ul>	<ul style="list-style-type: none"><li>● Quiz week 1</li></ul>	
2	<ul style="list-style-type: none"><li>● Quality attributes and QA scenarios</li><li>● Description and tactics for<ul style="list-style-type: none"><li>○ Performance</li><li>○ Availability</li><li>○ Security</li><li>○ Usability</li><li>○ Modifiability</li></ul></li><li>● Other quality attributes</li></ul>	<ul style="list-style-type: none"><li>● Student presentation</li><li>● Quiz week 2</li></ul>	
3	<ul style="list-style-type: none"><li>● Architecture views</li><li>● Module views that show module decomposition</li><li>● Module views that show module use</li><li>● Layered style</li><li>● MVC style</li><li>● Data model and domain model</li></ul>	<ul style="list-style-type: none"><li>● Student presentation</li><li>● Quiz week 3</li></ul>	
4	<ul style="list-style-type: none"><li>● Runtime (C&amp;C) views</li><li>● Pipe-and-filter style</li><li>● REST style</li><li>● Multi-tier style</li><li>● SOA style</li><li>● Peer-to-peer style</li><li>● Publish-subscribe</li></ul>	<ul style="list-style-type: none"><li>● Student presentation</li><li>● Quiz week 4</li></ul>	
5	<ul style="list-style-type: none"><li>● Deployment views</li><li>● Deployability</li><li>● Cloud computing</li><li>● Containerization</li><li>● Microservice style</li><li>● Blue-green deployment</li></ul>	<ul style="list-style-type: none"><li>● Student presentation</li><li>● Quiz week 5</li></ul>	

6	<ul style="list-style-type: none"><li>● Software interfaces</li><li>● Describing behavior<ul style="list-style-type: none"><li>○ UML sequence diagrams</li><li>○ BPMN and UML activity diagram</li><li>○ State machine diagram</li></ul></li><li>● Template for architecture views</li><li>● Architecture Decision Record (ADR)</li></ul>	<ul style="list-style-type: none"><li>● Student presentation</li><li>● Quiz week 6</li></ul>	
7	<ul style="list-style-type: none"><li>● Wrap-up</li><li>● Final exam</li></ul>	<ul style="list-style-type: none"><li>● Student presentation</li></ul>	

## References

[Bass01]	L. Bass, B. John, J. Kates. " <a href="#">Achieving Usability Through Software Architecture</a> ". CMU/SEI-2001-TR-005, March 2001.
[Bass13]	L. Bass, P. Clements, R. Kazman. <a href="#">Software Architecture in Practice, Third Edition</a> . Addison-Wesley, 2013.
[Bass21]	L. Bass, P. Clements, R. Kazman. <a href="#">Software Architecture in Practice, Fourth Edition</a> . Addison-Wesley, 2021.
[Bianco07]	P. Bianco, R. Kotermanski, P. Merson. " <a href="#">Evaluating a Service-Oriented Architecture</a> ". CMU/SEI-2007-TR-015, September 2007.
[Bianco11]	P. Bianco, G. Lewis, P. Merson, S. Simanta. " <a href="#">Architecting Service-Oriented Systems</a> ". CMU/SEI-2011-TN-008, August 2011.
[Buschmann96]	F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. <a href="#">Pattern-Oriented Software Architecture - A System of Patterns</a> . Wiley and Sons, 1996.
[Clements10]	Clements P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford. <a href="#">Documenting Software Architectures: Views and Beyond, Second Edition</a> . Addison-Wesley, 2010.
[Cockburn05]	Alistair Cockburn " <a href="#">Hexagonal Architecture</a> ." 2005.
[Costa15]	Bruno Costa, Paulo Pires, Paulo Merson. " <a href="#">Evaluating REST Architectures - Approach, Tooling and Guidelines</a> ". The Journal of Systems and Software. Elsevier, October 2015.
[DeRemer76]	Frank DeRemer and Hans H. Kron. " <a href="#">Programming-in-the-Large versus Programming-in-the-Small</a> ". IEEE Transactions on Software Engineering, June 1976.
[docker18]	" <a href="#">Docker Overview</a> ". Docker online documentation.
[Dijkstra68]	E. W. Dijkstra. " <a href="#">The structure of the THE multiprogramming system</a> ". Communications of the ACM, 1968.
[Erl07]	T. Erl. <a href="#">SOA Principles of Service Design</a> . Prentice Hall, July 2007.
[Erl09]	T. Erl. <a href="#">SOA Design Patterns</a> . Prentice Hall, January 2009.
[Evans03]	E. Evans. <a href="#">Domain-Driven Design: Tackling Complexity in the Heart of Software</a> . Addison-Wesley Professional, 2003.
[Fairbanks10]	G. Fairbanks. <a href="#">Just Enough Software Architecture</a> . Marshall & Brainerd, 2010.
[Fielding00]	R. Fielding. <a href="#">Architectural Styles and the Design of Network-Based Software Architectures</a> . PhD diss., University of California, Irvine, 2000.
[flow.ci16]	" <a href="#">Introduction to Containers: Concept, Pros and Cons, Orchestration, Docker, and Other Alternatives</a> ". Blog post, September 2016.

[Foote97]	B. Foote and Joseph Yoder. " <a href="#">Big Ball of Mud</a> ". PLoP'97.
[Fowler99]	M. Fowler. <a href="#">Refactoring: Improving the Design of Existing Code</a> . Addison-Wesley, June 1999.
[Fowler02]	M. Fowler. <a href="#">Patterns of Enterprise Application Architecture</a> . Addison-Wesley, November 2002.
[Fowler10]	M. Fowler. " <a href="#">Richardson Maturity Model - steps toward the glory of REST</a> ". March 2010.
[Gamma94]	E. Gamma et al. <a href="#">Design Patterns: Elements of Reusable Object-Oriented Software</a> . Addison-Wesley, 1994.
[Garlan94]	D. Garlan and M. Shaw. " <a href="#">An Introduction to Software Architecture</a> ". CMU-CS-94-166, January 1994.
[Helland18]	P. Helland. " <a href="#">Mind Your State for Your State of Mind</a> ". ACM Queue, volume 16 issue 3, July 2018.
[Hohpe03]	Hohpe, G. & Woolf, B. <a href="#">Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions</a> . Addison-Wesley, 2003.
[Hohpe16]	G. Hohpe, I. Ozkaya, O. Zimmermann. " <a href="#">The Software Architect's Role in the Digital Age</a> ". IEEE Software, November/December 2016.
[ISO 42010]	<a href="#">Systems and Software Engineering—Architectural Description International Standard</a> . ISO/IEC/IEEE 42010:2011(E). December 2011.
[Kazman00]	R. Kazman, M. Klein, P. Clements. " <a href="#">ATAM: Method for Architecture Evaluation</a> ". CMU/SEI-2000-TR-004. August 2000.
[Kazman03]	R. Kazman, L. O'Brien, C. Verhoef. " <a href="#">Architecture Reconstruction Guidelines, Third Edition</a> ". CMU/SEI-2002-TR-034. November 2003.
[Keeling17]	M. Keeling, J. Runde. " <a href="#">Architecture Decision Records in Action</a> ". Presentation at the SATURN Conference. May 2017.
[Kruchten95]	P. Kruchten. " <a href="#">The 4+1 View Model of Software Architecture</a> ". IEEE Software (Nov. 1995): 42-50.
[Kruchten03]	P. Kruchten. <a href="#">The Rational Unified Process: An Introduction (3rd Edition)</a> . Addison-Wesley, December 2003.
[Kruchten06]	P. Kruchten, H. Obbink & J. Stafford (2006). " <a href="#">The past, present and future of software architecture</a> ". IEEE SOFTWARE, March/April 2006.
[Lewis14]	Lewis, J.; & Fowler, M. " <a href="#">Microservices</a> ". 2014.
[Martin06]	R. C. Martin. <a href="#">Agile Principles, Patterns, and Practices in C#</a> . Pearson, July 2006.
[Martin08]	R. C. Martin. <a href="#">Clean Code: A Handbook of Agile Software Craftsmanship</a> . Pearson Education, August 2008.
[Medvidovic97]	N. Medvidovic, R.N. Taylor. " <a href="#">A Classification and Comparison Framework for Software Architecture Description Languages</a> ". Tech. Report UCI-ICS-97-02, University of California, Irvine, Feb. 1997

[Merson09]	P. Merson. " <a href="#">Data Model as an Architectural View</a> ". CMU/SEI-2009-TN-024. October 2009.
[Merson13a]	P. Merson and D. Saini. " <a href="#">Adventure Builder - Software Architecture Document (SAD)</a> ". SEI wiki, 2013.
[Merson13b]	P. Merson. " <a href="#">Ultimate Architecture Enforcement: Custom checks enforced at code-commit time</a> ". Proceedings of the SPLASH Conference, October 2013.
[Merson15a]	P. Merson. " <a href="#">Microservices Beyond the Hype: What You Gain and What You Lose</a> ". SEI Blog, November 2015.
[Merson15b]	P. Merson. " <a href="#">Defining Microservices</a> ". SEI Blog, November 2015.
[Merson16]	P. Merson. " <a href="#">Microservice design pattern</a> ". SOAPatterns.org, April 2016.
[Merson21]	P. Merson. " <a href="#">Principles for Microservice Design: Think IDEALS, Rather than SOLID</a> ". The InfoQ eMag Issue #91, February 2021.
[Meyer97]	B. Meyer. <a href="#">Object-Oriented Software Construction, 2nd Edition</a> . Prentice Hall, 1997.
[Millett15]	S. Millett & N. Tune. <a href="#">Patterns, Principles, and Practices of Domain-Driven Design</a> . Wrox, 2015.
[Millsap10]	C. Millsap. " <a href="#">Thinking Clearly About Performance, Part 2</a> ." Communications of the ACM, 53, 10 (October 2010): 39–45.
[Moussa19]	H. Moussa. " <a href="#">Business process visibility &amp; event-driven services</a> ". December 2019..
[NATO68]	P. Naur and B. Randell. " <a href="#">Software Engineering: Report of a conference sponsored by the NATO Science Committee</a> ", Garmisch, Germany, 7-11 Oct. 1968.
[Newman15]	S. Newman. " <a href="#">Pattern: Backends For Frontends</a> ". November 2015.
[Newman19]	S. Newman, <a href="#">Monolith to Microservices</a> . O'Reilly Media, November 2019.
[Newman21]	Newman, S. <a href="#">Building Microservices Second Edition</a> . O'Reilly Media, August 2021.
[Nygard11]	M. Nygard. " <a href="#">Documenting Architecture Decisions</a> ". Think Relevance blog, November 2011.
[Nygard18]	M. Nygard. <a href="#">Release It!, 2nd Edition</a> . O'Reilly Media, January 2018.
[O'Brien 03]	L. O'Brien, C. Stoermer. " <a href="#">Architecture Reconstruction Case Study</a> ". CMU/SEI-2003-TN-008. April 2003.
[Parnas72]	D. Parnas. " <a href="#">On the criteria to be used in decomposing systems into modules</a> ". Communications of the ACM, vol. 15, no. 12, December 1972, pp. 1053-1058.
[Parnas74]	D. Parnas. ' <a href="#">On a "buzzword": hierarchical structure</a> '. Proc IFIP Congress 74, 1974
[Parnas85]	D. Parnas, P. Clements, and D. Weiss. " <a href="#">The Modular Structure of Complex Systems</a> ". IEEE Transactions on Software Engineering, Vol. SE-11, No. 3, March 1985, pp. 259-266.
[Perry92]	D. E. Perry and A. L. Wolf. " <a href="#">Foundations for the study of software architecture</a> ". ACM SIGSOFT Software Engineering Notes, 17:pp. 40-52, October 1992.



[Platt02]	M. Platt. " <a href="#">Microsoft Architecture Overview</a> ". .NET Architecture Center, July 2002.
[Richardson16]	Richardson, C. " <a href="#">Database per Microservice pattern</a> ".
[Richardson18]	Richardson, C. <a href="#">Microservice Patterns</a> . Manning, 2018. (See also: <a href="#">microservices.io</a> .)
[Roberts18]	M. Roberts. " <a href="#">Serverless Architectures</a> ". May 2018.
[Rotem-Gal-Oz12]	Rotem-Gal-Oz, A. <a href="#">SOA Patterns</a> . Manning, 2012.
[Royce91]	W.E. Royce & W. Royce, "Software Architecture: Integrating Process and Technology." TRW Quest, 1991
[Ruka14]	A. Ruka. " <a href="#">My primer on Docker</a> ". December 2014.
[SATURN18]	" <a href="#">Panel: Death of the Architect</a> ". SATURN Conference, 2018.
[Shaw06]	M. Shaw and P. Clements. " <a href="#">The Golden Age of Software Architecture: A Comprehensive Survey</a> ". CMU-ISRI-06-101, February 2006.
[Shaw96]	M. Shaw and D. Garlan. <a href="#">Software Architecture: Perspectives on an Emerging Discipline</a> . Prentice Hall, 1996
[Smith03]	D. Smith and P. Merson. " <a href="#">Using Architecture Evaluation to Prepare a Large Web-Based System for Evolution</a> ". WSE 2003.
[SOA09]	" <a href="#">SOA Manifesto</a> ". 2009.
[Tyree05]	J. Tyree and A. Akerman. " <a href="#">Architecture Decisions: Demystifying Architecture</a> ". IEEE Software. March/April 2005.
[Vernon13]	V. Vernon. <a href="#">Implementing Domain-Driven Design</a> . Addison-Wesley Professional, February 2013.
[Wojcik06]	R. Wojcik, F. Bachmann, L. Bass, P. Clements, P. Merson, R. Nord, and W. Wood. " <a href="#">Attribute-Driven Design (ADD), Version 2.0</a> ". CMU/SEI-2006-TR-023, November 2006.
[Yoder19]	J. Yoder, A. Aguiar, P. Merson, H. Washizaki. " <a href="#">Deployment Patterns for Confidence</a> ". Asian PLoP. March 2019.
[Yoder20]	J. Yoder & P. Merson. " <a href="#">Strangler Patterns</a> ". PLoP. October 2020.
[Yogi17]	Y. Srikrishnan. <a href="#">The Death of the Architect</a> . DZone, 2017.

**Accommodations for Students Disabilities.** If you have a disability and have an accommodations letter from the Disability Resources office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I 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, I encourage you to contact them at [access@andrew.cmu.edu](mailto: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 me 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.

**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.

**We must treat every individual with respect.** We are diverse in many ways, and this diversity is fundamental to building and maintaining an equitable and inclusive campus community. Diversity can refer to multiple ways that we identify ourselves, including but not limited to race, color, national origin, language, sex, disability, age, sexual orientation, gender identity, religion, creed, ancestry, belief, veteran status, or genetic information. Each of these diverse identities, along with many others not mentioned here, shape the perspectives our students, faculty, and staff bring to our campus. We, at CMU, will work to promote diversity, equity and inclusion not only because diversity fuels excellence and innovation, but because we want to pursue justice. We acknowledge our imperfections while we also fully commit to the work, inside and outside of our classrooms, of building and sustaining a campus community that increasingly embraces these core values.

Each of us is responsible for creating a safer, more inclusive environment.

Unfortunately, incidents of bias or discrimination do occur, whether intentional or unintentional. They contribute to creating an unwelcoming environment for individuals and groups at the university. Therefore, the university encourages anyone who experiences or observes unfair or hostile treatment on the basis of identity to speak out for justice and support, within the moment of the incident or after the incident has passed. Anyone can share these experiences using the following resources:

- **Center for Student Diversity and Inclusion:** [csdi@andrew.cmu.edu](mailto:csdi@andrew.cmu.edu), (412) 268-2150
- **Report-It online anonymous reporting platform:** [reportit.net](http://reportit.net) username: *tartans*  
password: *plaid*

All reports will be documented and deliberated to determine if there should be any following actions. Regardless of incident type, the university will use all shared experiences to transform our campus climate to be more equitable and just.