

17655

Course Syllabus

Architectures for Software Systems Spring 2019 (17-655 and 17-755)

Course Objectives: Architectures for Software Systems aims to teach you how to design, understand, and evaluate systems at an architectural level of abstraction. By the end of the course you will be able to:

- Understand the influence of architectural drivers on software structures.
- Understand the technical, organizational, and business role of software architecture.
- Identify key architectural structures (styles, patterns, tactics, etc.).
- Understand the principles of good architectural documentation and presentation.
- Understand the impact that open source and third party components have on architectural designs.
- Generate architectural alternatives in a given context and choose among them.
- 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.

Course Organization:

Lectures: There will be two lectures weekly. Attendance at lectures is mandatory.

Recitations: To help you master key techniques and tools, provide guidance for assignments, and review the results of assignments, we will hold several recitations. Recitations will be held on Fridays. For some recitations, the class will be divided into smaller groups.

Computing: A personal computer or laptop is required for this course. For the 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, however, each of the assignments requiring coding can be completed with a simple text editor and command line prompt.

Course Meetings:

Class: Monday/Wednesday, 1:30PM - 2:50PM

Recitation: Fridays, 12:00PM – 1:20PM

Classroom: 300 South Craig Street, Room 265

Instructors:

David Garlan

garlan@cs.cmu.edu

WeH 4218

412-268-5056

Office hours by appointment

Assistant: Margaret Weigand (weigand@cs.cmu.edu)

Matthew Bass

mbass@cs.cmu.edu

300 SCR, Room 267

412-268-6163

Office hours by appointment

Teaching Assistants:

Sujata Telang

stelang@andrew.cmu.edu

300 SCR, Room 273

412-268-4644

Office hours: Fri 1:30-2:30pm

Cody Kinner

cbkinner@gmail.com

Office hours: Wed 3-4pm

CJ Zhang

changjiz@andrew.cmu.edu

Office hours: Mon 3-4pm

Grading Policies:

- Late assignments are penalized 10% per day
- Lowest 2 quizzes are dropped
- No late or makeup quizzes
- Instructor discretion adjustment based on participation, effort, and peer review



Assignment Submission Policies:






- All submitted assignments should be named *docx*(where *AndrewID* = your andrew id and *AssignmentNumber* = the assignment number)
- The time uploaded will be counted as the submission time
- All assignments are to be submitted to Canvas
- All assignments will be in either Microsoft Word or pdf format


Required Texts:

- *Software Architecture in Practice, Third Edition*, by Bass, Clements, Kazman, Addison-Wesley 2013 [BCK13].
- *Architecting Software Intensive Systems: A Practitioner's Guide*, Lattanze, Auerbach 2008 [ASIS08].
- *Documenting Software Architectures: Views and Beyond, Second Edition*, by Clements, et al. Addison-Wesley 2011 [C+11].
- We will also use a collection of supplementary readings, which will be available through the course web site.

Schedule

Lecture	Date	Major Topic	Lecture/Recitation	Assignments	Readings
1	Mon Jan 14	Architectures Overview	Course Introduction 		
2	Wed Jan 16		Business Goals and Architectural Drivers 		<ul style="list-style-type: none">• Lecture 2 - Exercise• ASIS ch 1• BCK ch 3
	Fri Jan 18		No Recitation		

	Mon Jan 20		No Class Martin Luther King Day		
3	Wed Jan 23		Architecture and Lifecycles 	A1 Assigned	<ul style="list-style-type: none"> • BCK ch 4
R1	Fri Jan 25		Architecture Drivers	Quiz	
4	Mon Jan 28		Documenting Architecture 		<ul style="list-style-type: none"> • Lecture 4 - Exercise • C+11 Prolog • C+11 ch 9
5	Wed Jan 30	Quality Attributes	Performance and Performance Tactics 		<ul style="list-style-type: none"> • BCK ch 8
R2	Fri Feb 1		Architectural Drivers	Quiz	
6	Mon Feb 4		Modifiability and Modifiability Tactics 	A1 Due	<ul style="list-style-type: none"> • BCK ch 7
7	Wed Feb 6		Availability and Availability Tactics 	A2 Assigned	<ul style="list-style-type: none"> • BCK ch 5
R3	Fri Feb 8			Quiz	<ul style="list-style-type: none"> • ABC Mobile Security

8	Mon Feb 11		Extensibility/Scalability and associated Tactics 		<ul style="list-style-type: none"> • GGL03 • Ham07
9	Wed Feb 13		Security and Security Tactics 		<ul style="list-style-type: none"> • BCK ch 9 • Saltzer et al. 
R4	Fri Feb 15			Quiz	
10	Mon Feb 18	Architectural Patterns	Dataflow Systems 		<ul style="list-style-type: none"> • BCK13 pps 215-216, 232- • Lat99
11	Wed Feb 20		Event Systems 		<ul style="list-style-type: none"> • Hoh06
R5	Fri Feb 22			Quiz	
12	Mon Feb 25		Call-and-Return Patterns 		<ul style="list-style-type: none"> • BCK13 pps 217-226; 235- • Par72
13	Wed Feb 27		Client-Server and Tiered Systems 	A2 Due A3 Assigned	<ul style="list-style-type: none"> • L13 Tiered Exercise • Intel01 • MS Client-Server • MS N-Tiered
R6	Fri Mar 1			Quiz	

14	Mon Mar 4		Middleware		<ul style="list-style-type: none"> • Bak14 • https://www.computerworld.com/news/2018/03/04/2018-03-04-the-necessary-evil-enterprise-middleware.html (Links to an external site.)
15	Wed Mar 6		Data Centric Systems		<ul style="list-style-type: none"> • Bur11b • Cor12 • Nak08
R7	Fri Mar 7			No Recitation Mid-Semester Break	
	Mon Mar 11 — Wed Mar 13	Spring Break			
16	Mon Mar 18		Embedded and Cyber Physical Systems		<ul style="list-style-type: none"> • GLT03 • Lee06
17	Wed Mar 20	Architecture to Implementation	Architecture Design In A Project Environment		<ul style="list-style-type: none"> • ASIS Chap 5
R8	Fri Mar 22			Quiz	<ul style="list-style-type: none"> • SJH15
18	Mon Mar 25		Architecture Design In An Agile Environment		<ul style="list-style-type: none"> • Cli96

19	Wed Mar 27		Implementing An Architecture	A3 Due	<ul style="list-style-type: none"> • SJH15
R9	Fri Mar 29			Quiz Initial Deliverable for Final Project	
20	Mon Apr 1		Architecture Reconstruction		<ul style="list-style-type: none"> • Reconstruction Exercise • A4 Assigned
21	Wed Apr 3		Development Operations		<ul style="list-style-type: none"> • IEEE Column
R10	Fri Apr 5			Quiz	
22	Mon Apr 8	Architectural Methods	Using UML to Document Architectures		<ul style="list-style-type: none"> • C+11 Appendix A, pps 43
23	Wed Apr 10		Architecture Evaluation		
R11	Fri Apr 12			Quiz	
24	Mon Apr 15		TBD		<ul style="list-style-type: none"> • ASIS08 ch 7 • Lat12

25	Wed Apr 17	Emerging Topics	Microservices		<ul style="list-style-type: none"> • J+18
R12	Fri Apr 19			Quiz Final Project Check in	
26	Mon Apr 22		Formal Methods and Architecture	A4 Due	<ul style="list-style-type: none"> • GS06 • SG03 • Optional: http://www.cs.cmu.edu/~rwh/21-448/
27	Wed Apr 24		Architecture Reuse and Product Lines		<ul style="list-style-type: none"> • BCK13 Ch 25 • Bos09 • GAO95
R13	Fri Apr 26			Quiz	
28	Mon Apr 29		Machine Learning for the Software Architect		<ul style="list-style-type: none"> • https://livebook.manning.com/livebook/systems/chapter-1/v-11/1 (Link to an external site.) • https://keon.io/deep-q-learning/ (Link to an external site.) • Optional: https://becomingascientist.com/dqn-df57e8ff3b26 (Links to an external site.)
29	Wed May 1		Future Directions		<ul style="list-style-type: none"> • GS07 • GCSS04 • Skim SG03 • Optional: Gar14
R14	Fri			Quiz	

	May 3				
30	Mon May 6	Finals Day 1		Final Project Due	
31	Wed May 8	Finals Day 2			

References

- ASIS *Architecting Software Intensive Systems: A Practitioner's Guide.*
Lattanze. Auerbach 2008.
- Bak14 Middleware. David E. Bakken. 2014
- BCK *Software Architecture in Practice, Third Edition.* Bass, Clements,
Kazman, Addison-Wesley 2013
- Bos09 From Software Product Lines to Software Ecosystems. Jan Bosch.
- Bur11b From Login, NoSQL. Greg Burd, October 2011.
- C+11 *Documenting Software Architectures: Views and Beyond, Second
Edition.* Clements, et al. Addison-Wesley 2011.
- ClI96 The Pros and Cons of Adopting and Applying Design Patterns in the
Real World. Marshal Cline. 1996
- Cor12 Collaborating Software, Blackboard and Multi-agent Systems & Future.
Daniel D. Corkill.
- GAO95 Architectural Mismatch or, Why it's hard to build systems out of existing
parts. Garlan, Allen, Ockerbloom.

- Gar14 Software Architecture: A Travelogue. David Garlan, ACM978-1-4503-2865-4/14/05.
- GCSS04 Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure. Garlan, Cheng, Schmerl, Steenkiste. *IEEE Computer* 0018-9162/04/2004.
- GGL03 The Google File System. Ghemawat, Gobioff, Leung. SOSP 03 October 19-22, 2003 ACM 1-58113-757-5/03/0010.
- GLT03 Embedded Software Engineering: The State of the Practice. Graaf, Lormans, Toetenel. *IEEE Software*, 0740-7459/03.
- GS06 Architecture-driven Modeling and Analysis. David Garlan and Bradley Schmerl.
- GS07 The RADAR Architecture for Personal Cognitive Assistance. David Garlan and Bradley Schmerl.. *International Journal of Software Engineering and Knowledge Engineering*, Vol. 17(2), April 2007.
- Ham07 On Designing and Deploying Internet-Scale Services. James Hamilton, 21stLarge Installation System Administration Conference (LISA07).
- Hoh06 Programming Without a Call Stack – Event-driven Architectures. Hohpe, 2006.
- IEEE Column The Software Architect and DevOps.
- Intel01 N-tier Architecture Improves Scalability, Availability, and Ease of Integration. Microsoft, “Client-Server Pattern.” <https://msdn.microsoft.com/en-us/library/ee658117.aspx#NTier3TierStyle>.
- J+18 Microservices, The Journey So Far and Challenges Ahead. Jamshidi, Pahl, Mendonca, Lewis, Tilkov. *IEEE Software* 0740-7459/18, 2018.
- Lat99 Case Study: An Early Exploitation of Product Lines. Anthony Lattanze.

- Lee06 Cyber-Physical Systems – Are Computing Foundations Adequate?
Edward Lee. NSF Workshop on Cyber-Physical Systems, October 2006.
- MS N-Tiered [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658117\(v=pandp.10\)#NTier3TierStyle](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658117(v=pandp.10)#NTier3TierStyle)
- Nak08 Bitcoin: A Peer-to-Peer Electronic Cash System. Nakamoto. 2008.
- Par72 On the Criteria for Decomposing Systems into Modules. *Comm. Of the ACM*, 15(12):1053-1058, December 1972.
- SS75 The Protection of Information in Computer Systems. Saltzer and Schroeder. MIT 1975.
- SG03 The Aura Software Architecture: an Infrastructure for Ubiquitous Computing. Joao Sousa and David Garlan. CMU-CS-03-183, August 2003.
- SJH15 Impact of Object Oriented Design Patterns on Software Development. Subburaj, Jekese, Hwata. *International Journal of Scientific & Engineering Research*, Volume 6, Issue 2, February 2015.