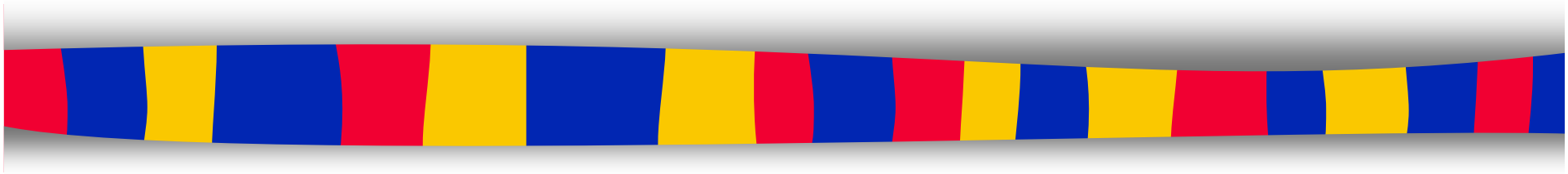
CISC 322

Software Architecture

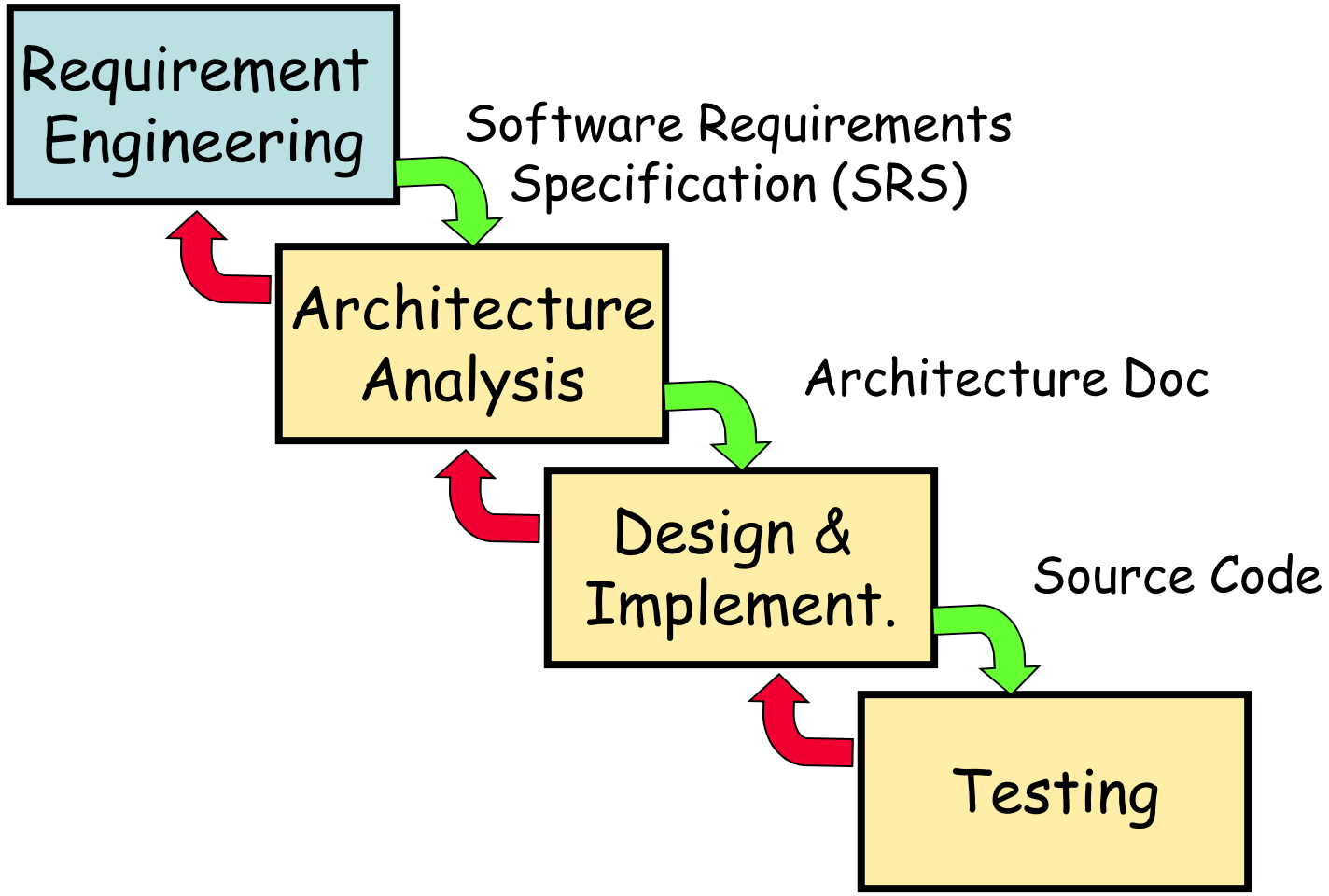


Lecture 01:

Introduction and Admin

Ahmed E. Hassan

Waterfall Development Process



Software Architecture (IEEE Definition)

- *Architecture is the fundamental **organization** of a **system** embodied in its **components**, their **relationships** to each other, and to the **environment**, and the principles guiding its design and evolution. [IEEE 1471]*

IEEE Definitions

- *A **system** is a collection of components organized to accomplish a specific function or set of functions. The term system encompasses individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, whole enterprises, and other aggregations of interest. A system exists to fulfill one or more **missions** in its environment. [IEEE 1471]*

IEEE Definitions

- *The **environment**, or context, determines the setting and circumstances of developmental, operational, political, and other influences upon that system. [IEEE 1471]*
- *A **mission** is a use or operation for which a system is intended by one or more **stakeholders** to meet some set of objectives. [IEEE 1471]*
- *A **stakeholder** is an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system. [IEEE 1471]*

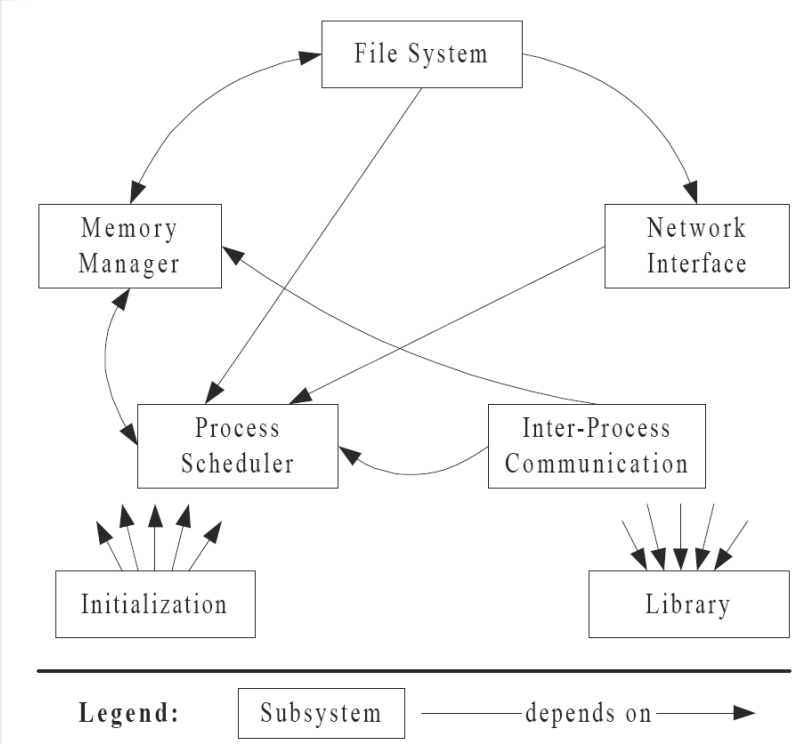
Software Architecture (Kruchten)

- *An architecture is the **set of significant decisions** about the organization of a software system, the selection of **structural elements** and their interfaces by which the system is composed, together with their **behavior** as specified in the collaborations among those elements, the **composition** of these elements into progressively larger subsystems, and the **architectural style** that guides this organization -- these elements and their interfaces, their collaborations, and their composition.*

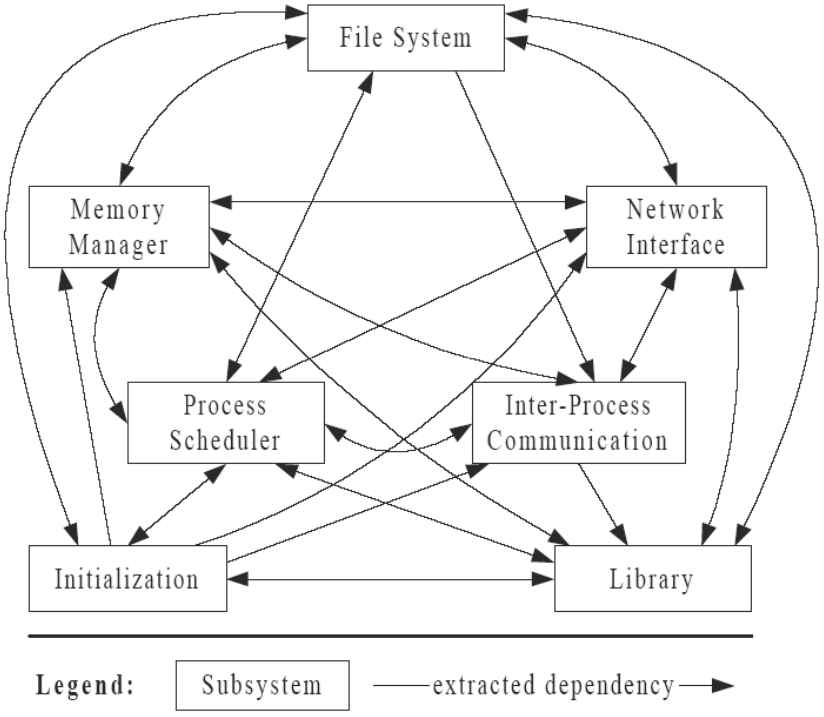
Terminology

- **System Architecture**
  - Structure: Several computers, networks, data bases, etc. connected together
  - *Analogy: Plan of city*
- **Conceptual Software Architecture**
  - Abstract structure: Large piece of software with many parts and interconnections
  - *Analogy: Blueprint of house*
- **Concrete Software Architecture**
  - Actual structure: Large piece of software with many parts
  - *Analogy: Actual structure of house*

Linux Architecture



Conceptual Architecture



Concrete Architecture

Architecture vs. Design

- **Architecture**
  - Structure of system (components and connectors)
  - High level and hard to change (better get it right!)
  - Concerned with technical and non technical requirements (e.g., Security, Legal, Outsourcing)
  - Makes sense for systems with MLOCs
  - Very early in life cycle
- **Design**
  - Inner structure of the components
  - Low level (information hidding and interfaces help it change)
  - Mostly technical concerns
  - Makes sense for systems with KLOCs
  - Late in life cycle

Understanding Software Architecture

- **Live Architecture**
  - Is in head(s) of software developer(s), the "software architect"
  - May be abstract or mostly concrete
  - Is a "mental model", "wetware"; may be fuzzy, inaccurate, incomplete, incorrect …
- **Complexity**
  - Architecture simplifies the system, by concentrating on structure, not content or semantics
  - Cognitive complexity: how hard to understand or visualize
- **Reverse Engineering**
  - Extraction of design (or architecture) from implementation and from developers
  - "Design recovery"

Course Scope

- Exposes you to the challenges in developing large and ultra large software systems
- Learn various concepts related to large scale software development
  - Architectural views
  - Architecture evaluation methods
  - Social architecture (conway's law)
  - Effort estimation techniques
  - Software evolution and software aging
  - Team leadership
- Study the architecture of a Large software system (*Google Chrome*)

Course Format

- Three slots:
  - Wednesday 10:00 – 11:30 JEF-319
  - Thursday 12:30 – 2:30 JEF-102
  - Friday 8:30 – 10:00 JEF-319
- You **MUST** attend all scheduled slots
  - Early in the term, extra lectures will be given in these slots to ramp up on the project details
  - Later in the term, you will have more time for meetings and discussions related to your project

Course Staff and Web Page

- Lecturer:
  - Ahmed E. Hassan, 156 Barrie Street, ahmed@cs.queensu.ca
  - Office Hours: 11:30-12:30 Wednesdays & by appointment
- TA:
  - Thanh H. Nguyen, ThanhNguyen@cs.queensu.ca
- Course Webpage:
  - http://www.cs.queensu.ca/~cisc322/
- Course Email:
  - cisc322@cs.queensu.ca
- Send emails from your queen's email account, otherwise likely to be flagged as spam
- Put "CISC322" in subject to go around spam filters

Course Expectations

- Read assigned readings
- Attend lectures and participate in discussions
- Bring your ideas and concerns to class
- Work effectively in a group setting (group members will evaluate each other)
- Learn how to use the tools and understand your project *very well*
- Hand in your deliverables on time

Evaluation

- Midterm (**15 Oct 2009**) **10%**
- Final **25%**
- Group Project **65%**

- **NOTE:**
  - You **HAVE** to pass the Midterm + Final to pass the course
  - You **HAVE** to pass the project to pass the course

Course Project

- A0: Create webpage forChrome
- A1: Describe the conceptual architecture
- A2: Recover the concrete architecture and compare to conceptual
- A3: Propose an enhancement and propose and compare 2 designs/implementation plans

Project Mark Breakdown

1.	Group List(4 Members per group)+Links(**A0**)	3%	**1 Oct 2009**
2.	Conceptual Architecture Presentation	7%	**21 Oct 2009**
3.	Conceptual Architecture (**A1**) [15 pgs]	10%	**23 Oct 2009**
4.	Concrete Architecture Presentation	7%	**11 Nov 2009**
5.	Concrete Architecture (**A2**) [15 pgs]	15%	**13 Nov 2009**
6.	Architecture Enhancement Presentation	7%	**2 Dec 2009**
7.	Architecture Enhancement (**A3**) [15 pgs]	16%	**4 Dec 2009**

Students must participate in all project presentations; missing the presentation slot for a deliverable will result in a 25% reduction in your mark for that presentation.

Website

- You need to update a group website throughout the term
- Website should be up by **1 Oct 2009**
  - Worth 3% of your mark
  - If not kept up-to-date you lose the 3%

Peer Reviews

- All members should receive same marks for project, however to account for that lumpiness, we have peer reviews
  - You can assign each member (including yourself) a grade
  - You have 5 * N + 1 marks, where N is size of group
- Peer reviews are sent 24 hours ***after*** each large deliverable:
  - A1 – 23 Oct 2009 + 24hrs
  - A2 – 13 Nov 2009 + 24hrs
  - A3 – 4 Dec 2009 + 24hrs
- YOUR mark depends on the reviews being sent in on time! (**25% off if delayed**).
- Reviews are submitted on WebCT with subject: “Peer Review for Group ##”

Lateness Policy for All Course Deliverables

- For all deliverables:
  - Hand hard copy at the beginning of class or earlier to instructor or TA
  - Submit online at WebCT

NO LATE DELIVERABLES!!

Academic Integrity and Cheating

- Cheating, plagiarism and other forms of academic fraud are taken very seriously by the University, the Faculty, and the teaching staff.
- Examples:
  - Submitting the work of another person as your original work
  - Incorporating others work in your work and not attributing it
  - It is permitted and encouraged to discuss projects with your peers on the whiteboard but **NOT** permitted to copy their solutions as they talk to you. Both parties would be penalized

Course Text

- There is no required text book for the course
- Lecture slides, papers, online books
- Additional online readings assigned for case studies
- Midterm and final will cover assigned readings and topics covered in class

Working in Groups and Choosing a Group

- Group Size : 4-5
- Understand the work habits and goals of your group members:
  - Night person
  - Start early
  - Laid back
  - Best project ever
  - Morning person
  - Start at last minute
  - Perfectionist
  - Reasonable mark
- Identify members with good English and communication skills

Asking Questions

- Ask me or TA (email, office hours)
- Ask in class
- Discuss with your classmates or group members
- Ask on the CISC 322 forum on WebCT

The Software Pyramid

Software programming is the iconic job of the Information Age, but not all programmers are created equal. Here's the breakdown of software jobs and their prospects:

1 ARCHITECTS A few thousand tech visionaries sketch out entire systems to handle complex jobs. Adam Bosworth, for example, is the chief architect at BEA Systems.

PAY $150,000 to $250,000.

OUTLOOK Outsourcing is a nonissue.

2 RESEARCHERS They're key to innovation, which is crucial for the U.S. But there are only about 25,000 in the country, many in academia, where tenure trumps pay.

PAY $50,000 in academia to $195,000 in private sector.

OUTLOOK Prospects should brighten somewhat with the economy, but these jobs can move offshore, too.

3 CONSULTANTS Business-savvy consultants advise corporations about their technology needs, help them install new software, and create new applications from scratch.

PAY $72,000 to $200,000.

OUTLOOK Still bright for Americans. U.S. customers want face time with consultants.

4 PROJECT MANAGERS Crucial cogs in global software factories. They coordinate the work of teams in different countries and time zones and provide dependable products on schedule.

PAY $96,000 to $130,000.

OUTLOOK Good managers can write their own tickets. Pay has jumped 14.3% in the past two years.

5 BUSINESS ANALYSTS Go-betweens. About 100,000 analysts figure out what a business needs and turn it into a spec sheet for programmers. It's a key role now since the company and its programmers are often apart.

PAY $52,000 to $90,000.

OUTLOOK A relatively safe haven for programmers—if they have communications skills and a grip on business.

6 BASIC PROGRAMMERS The foot soldiers in the information economy, they write the code for applications and update and test them. Numbering about 1 million, they are one-third of all U.S. software engineers and programmers.

PAY Has tumbled 15% since 2002. Now $52,000 to $81,000.

OUTLOOK Watch out. Many of these jobs can be done anywhere. Forrester predicts 18% of them will be offshore within six years.

Data: Forrester Research, Foote Partners, Kennedy Information Inc., *BusinessWeek*

BusinessWeek March, 2004

More Terminology

- **Architectural Style**
  - Form of structure, e.g.,
    - "Pipes" between components, or
    - "Layered" system, or
    - "Bulletin board" system
  - *Analogy: Style of a building*
- **Reference Architecture**
  - General architecture for an application domain
  - *Example: Common structure for compilers or for operating systems*
- **Product Line Architecture (PLA)**
  - Architecture for a line of similar software products
  - *Example: Software structure for a family of computer games*