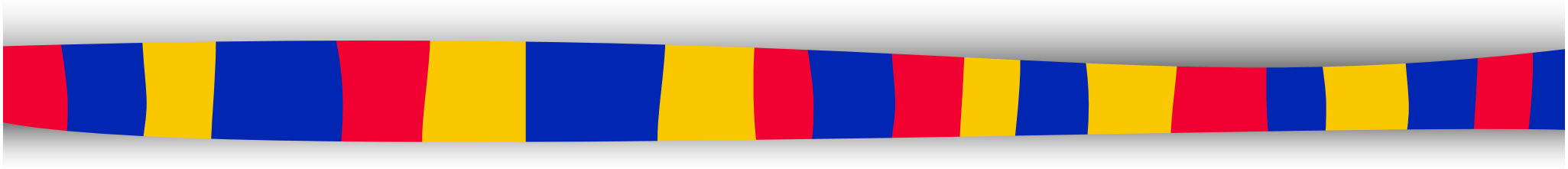
CISC 326

Game Architecture

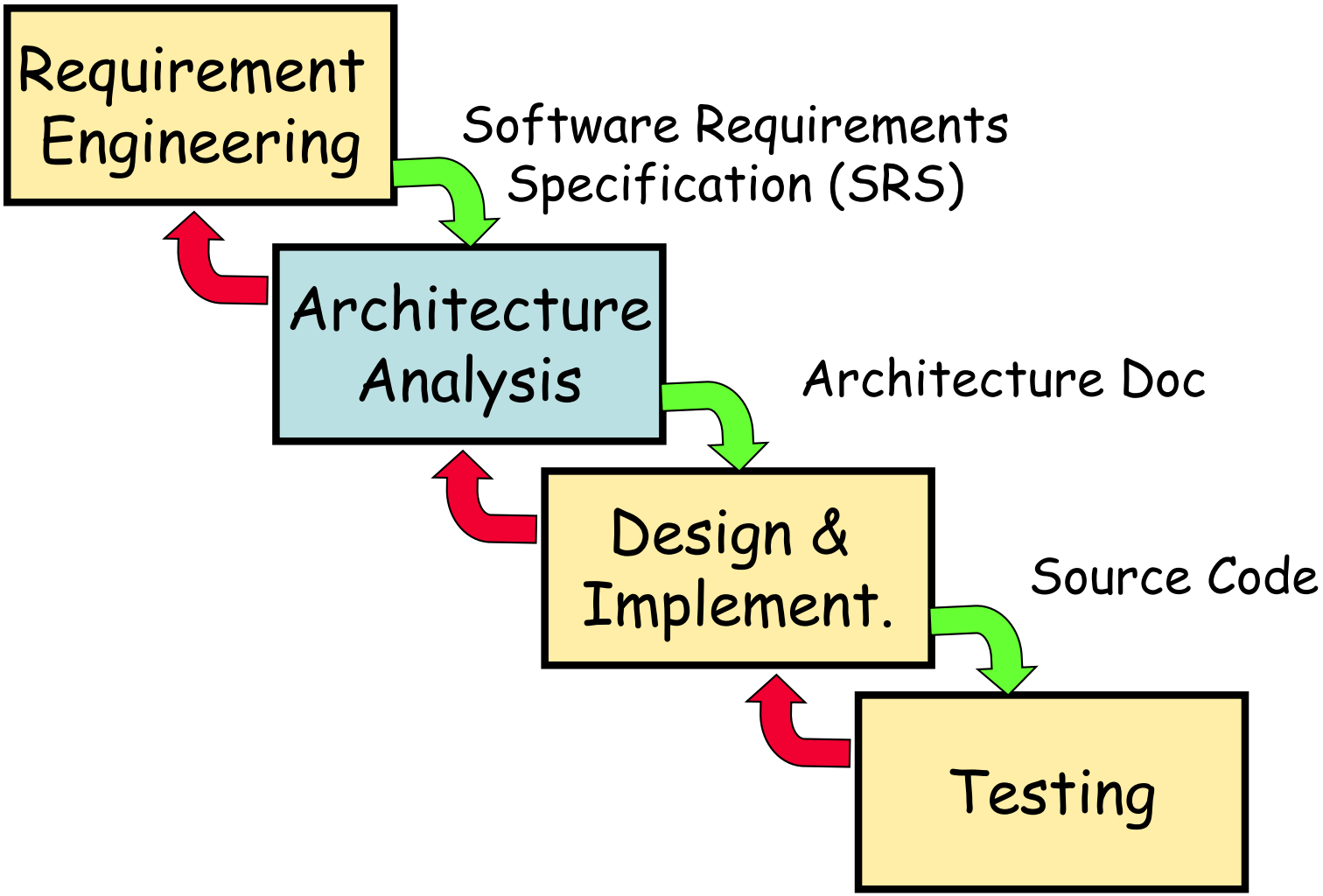


Module 01:

Introduction, Syllabus, 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]*

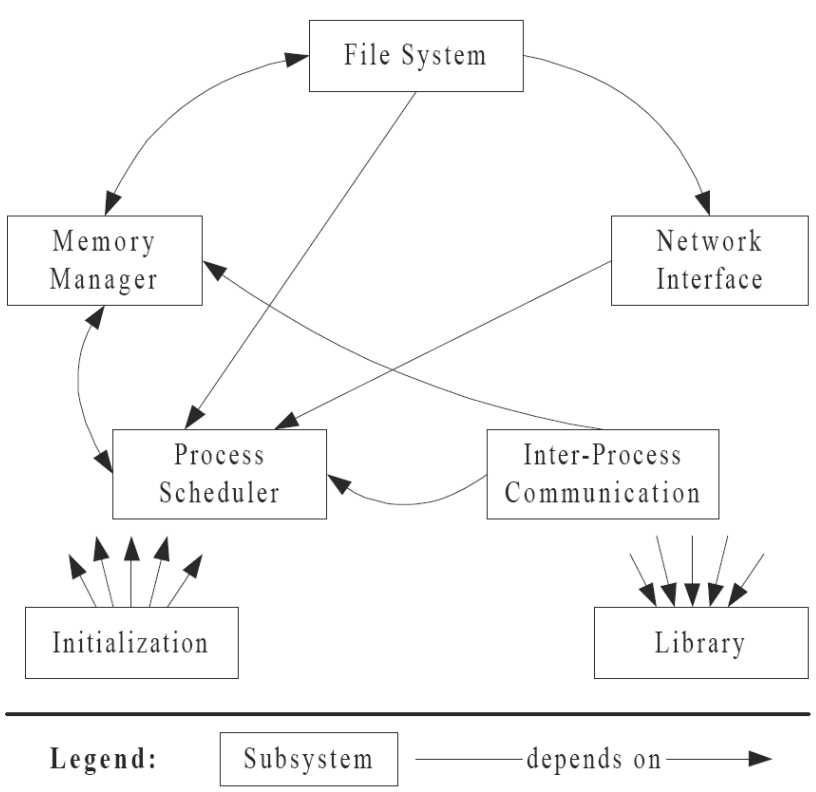
Software Architecture (Kruchten)

- *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.*

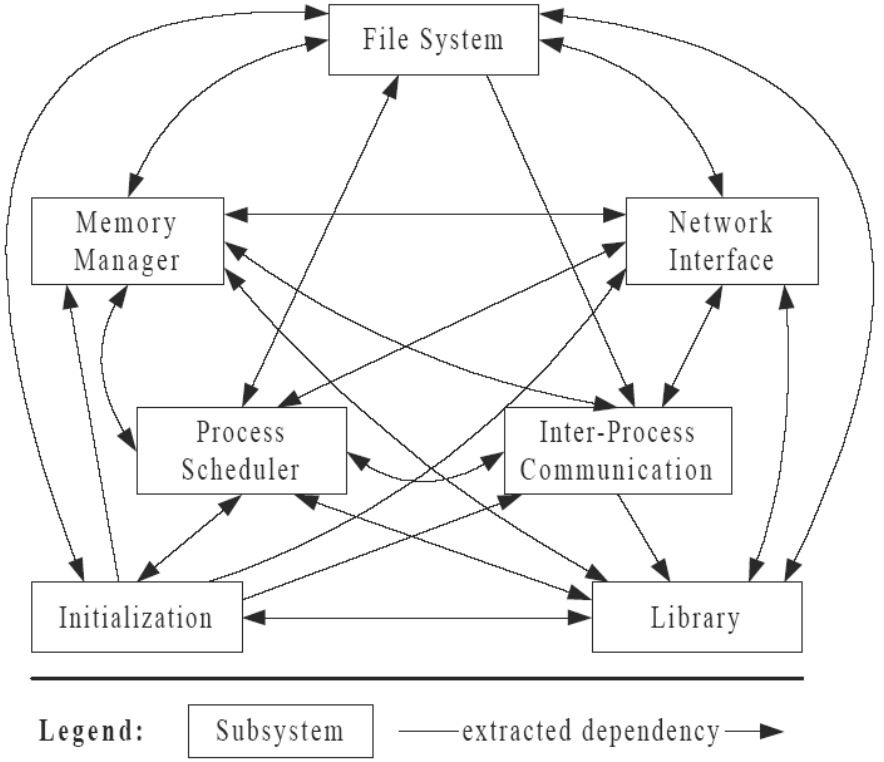
Terminology

- **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, and games
- 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 game

2015 Project

2016 Project:
SuperTuxKart

This Year's Project:
SuperTux

http://supertux.org

Course Format

- **Three slots:**
  - Monday 1:30PM to 2:30PM --- STIRLING AUD
  - Wednesday 12:30PM to 1:30PM --- STIRLING AUD
  - Friday 11:30AM to 2:30PM --- CHERNOFF AUD
- 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

NO COURSE CONFLICTS ARE PERMITTED

Course Staff and Web Page

- Lecturer:
  - Ahmed E. Hassan, 133 Princess Street, ahmed@cs
  - Office Hours: by appointment
- TA:
  - Dayi Lin (Head TA)
  - Siyi Emily Bao
  - Jake Gibbons.
  - Heng Li
- Course Webpage:
  - http://www.cs.queensu.ca/~cisc326/
- Course Email:
  - cisc326@cs.queensu.ca
- Send emails from your queen's email account, otherwise likely to be flagged as spam
- Put **[CISC326]** 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

- Quiz 1 (**13 Oct 2017**) **15%**
- Quiz 2 (**17 Nov 2017**) **20%**
- Group Project **65%**

- **NOTE:**
  - You **HAVE** to pass **EACH** quiz to pass the course
  - You **HAVE** to pass the project to pass the course
  - Both Quizes will be done during class\tutorial time

Course Project

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

Project Mark Breakdown

1.	Submit Group of 6 (otherwise random group)		**18 Sep 2017**
2.	Group Website (**A0**)	3%	**2 Oct 2017**
3.	Conceptual Architecture Presentation	7%	**16 Oct 2017**
4.	Conceptual Architecture (**A1**) [15 pgs]	10%	**23 Oct 2017**
5.	Concrete Architecture Presentation	7%	**6 Nov 2017**
6.	Concrete Architecture (**A2**) [15 pgs]	15%	**13 Nov 2017**
7.	Architecture Enhancement Presentation	7%	**27 Nov 2017**
8.	Architecture Enhancement (**A3**) [15 pgs]	16%	**1 Dec 2017**

All deadlines are 8 AM, except #8 (11:59 PM)

You must attend the whole class at which your group presents not just your group presentation.

Google Calendar with deadlines here: https://goo.gl/tyj6RC

Website

- You need to update your group website throughout the term
- Website should be up by **2 Oct 2017 (URL submitted via onQ)**
  - Worth 3% of your mark
  - If not kept up-to-date you lose the 3%

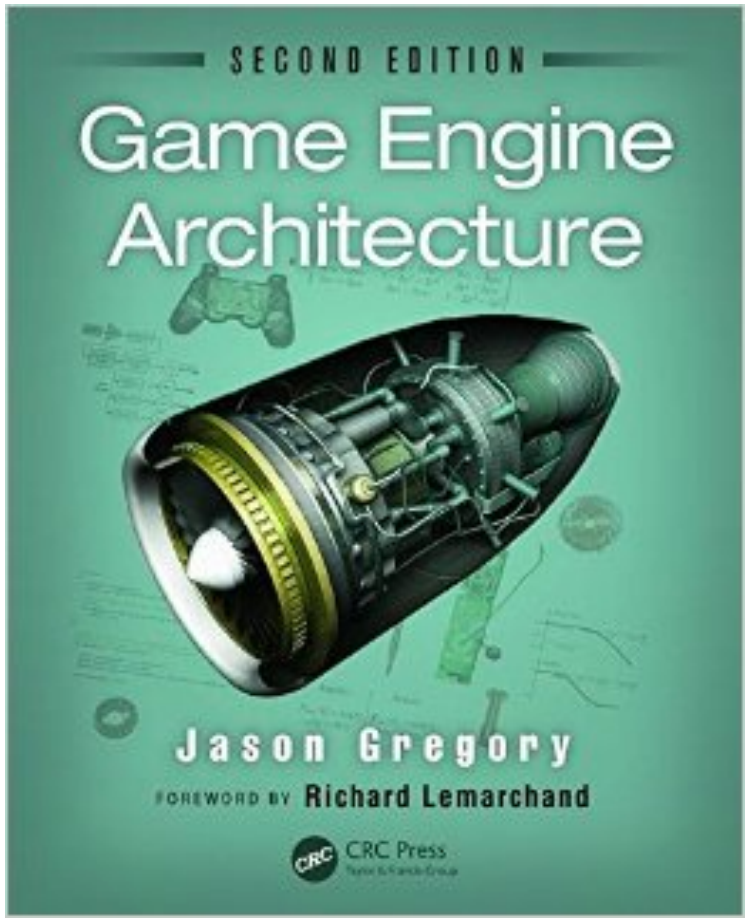
Peer Reviews

- All members should receive same marks for project, however to account for that lumpiness, we have peer reviews
  - You must assign each member (including yourself) a grade
  - You have 5 * N + 1 marks, where N is size of group
- Peer reviews are due 24 hours ***after*** each large deliverable
- YOUR mark depends on the reviews being sent in on time! (**25% off if missed deadline, 100% off if not submitted within four days).**
- **Reviews are a TEXT FILE submitted via onQ (if not a text file, it is not submitted) NO WORD files!**

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

Recommended Course Book



Working in Groups and Choosing a Group

- **Group Size : 6**
- Understand the work habits and goals of your group members:
  - Night person
  - Early starter
  - Laid back
  - Best project ever
  - Morning person
  - Last minute starter
  - Perfectionist
  - Reasonable mark
- Identify members with good English and communication skills

Asking Questions

- Ask me or TAs (email, office hours)
- Ask in class
- Discuss with your classmates or group members

Grading Method (numbers in, letters out)

- All components of this course will receive numerical percentage marks. The final grade you receive for the course will be derived by converting your numerical course average to a letter grade according to Queen's Official Grade Conversion Scale

Queen's Official Grade Conversion Scale

Grade	Numerical Course Average (Range)
A+	90-100
A	85-89
A-	80-84
B+	77-79
B	73-76
B-	70-72
C+	67-69
C	63-66
C-	60-62
D+	57-59
D	53-56
D-	50-52
F	49 and below

Lateness Policy for All Course Deliverables

- For all deliverables:
  - Submit online via onQ on due date/time

NO LATE DELIVERABLES!!

Academic Integrity

- 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

Academic Integrity

- Academic Integrity is constituted by the six core fundamental values of honesty, trust, fairness, Respect, responsibility and courage (see www.academicintegrity.org). These values are central to the building, nurturing and sustaining of an academic community in which all members of the community will thrive. Adherence to the values expressed through academic integrity forms a foundation for the "freedom of inquiry and exchange of ideas" essential to the intellectual life of the University (see the Senate Report on Principles and Priorities http://www.queensu.ca/secretariat/policies/senate/report-principles-and-priorities).
- Students are responsible for familiarizing themselves with the regulations concerning academic integrity and for ensuring that their assignments conform to the principles of academic integrity. Information on academic integrity is available in the Arts and Science Calendar (see Academic Regulation 1 http://www.queensu.ca/artsci/academic-calendars/regulations/academicregulations/regulation-1), on the Arts and Science website (see http://www.queensu.ca/artsci/academics/undergraduate/academic-integrity), and from the instructor of this course.
- Departures from academic integrity include plagiarism, use of unauthorized materials, facilitation, forgery and falsification, and are antithetical to the development of an academic community at Queen's. Given the seriousness of these matters, actions which contravene the regulation on academic integrity carry sanctions that can range from a warning or the loss of grades on an assignment to the failure of a course to a requirement to withdraw from the university.

Turnitin Statement

- Queen's University has partnered with the third-party application Turnitin to help maintain our standards of excellence in academic integrity.
- Turnitin is a suite of tools that provide instructors with information about the authenticity of submitted work and facilitates the process of grading.
- Submitted files are compared against an extensive database of content, and Turnitin produces a similarity report and a similarity score for each assignment. A similarity score is the percentage of a document that is similar to content held within the database.
- Turnitin does not determine if an instance of plagiarism has occurred. Instead, it gives instructors the information they need to determine the authenticity of work as a part of a larger process

Special Accommodations

- Queen's University is committed to achieving full accessibility for persons with disabilities. Part of this commitment includes arranging academic accommodations for students with disabilities to ensure they have an equitable opportunity to participate in all of their academic activities. If you are a student with a disability and think you may need accommodations, you are strongly encouraged to contact Student Wellness Services (SWS) and register as early as possible. For more information, including important deadlines, please visit the Student Wellness website at: http://www.queensu.ca/studentwellness/accessibility-services/

Accommodation Requests need to be submitted AT LEAST TWO WEEKS before needed deadline

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*