4+1 Model of Architecture:

**Scenarios**

**Physical View:**

Systems engineers, Topology, Communications

Shows the system hardware and how software components are distributed across the processors in the system.

**Development View:**

Programmer’s software management

Shows how the software is decomposed for development

**Process View:**

Integrators, Performance, Scalability

Shows how, at runtime, the system is composed of interacting processes

**Logical View:**

End user functionality

Shows key abstractions as objects or object classes

Context Models



Lines connect actors to use cases

Structural Models

Sequence Diagrams

Use case Models

Lines represent association

1, 1..\*, 1..4, etc. Numbers represent how many objects are involved in the association

Boxes represent Classes

Alt

Alt box represent alternate interactions

Arrows represent interactions

Boxes represent Systems

Figures represent actors

Figures represent actors

Figures represent actors

Arrows represent the flow of work

Filled circle inside another circle represents the end of a process

Filled circle represents the start of a process

Rounded boxes represent activities

Boxes represent Systems

**1.Lec:** Increasing role of software in all systems. **Increase of heterogeneous, networked, distributed, and interdependent systems.** Implications of failure in these systems. **Pervasive collection, manipulation, and use of digital data and information.** Importance of software and intellectual property to the economy. **\*\*\* Define SE.**

**2.KP:** SW processes are the activities involved in producing a software system, and their models are abstract representations of the processes. **General process models describe the organization of SW processes. These include the Waterfall model, Incremental Development, and reusable component configuration and integration.** Requirements Engineering is the process of developing a SW specification. **Design and implementation processes transform requirements into executable SW systems.** SW validation makes sure the system follows the specification. **Processes should cope with change and be structured for iterative development.** Process improvement is improving existing SW to improve quality, lower costs, or reduce time. It is a cyclic process involving measurement, analysis, and change.

**4.Lec:** Kabayashi Maru problem: Time, Cost, Quality, scope  
\*\*\* **WaterFall:** Reqs. definition>**System and SW design>** Implementation and unit testing>Integration and system testing>Operation and maintenance. \*\*\* **Agile Manifesto:** Individuals and interactions over process and tools. **Working software over comprehensive documentation.** Customer collaboration over contract negotiation. **Responding to change over following a plan.** \*\*\* Scrum terms: **Dev Team,** Product owner, **Scrum Master: In charge of guiding team using scrum,** Scrum: A daily meeting of the team to review progress and prioritize work. **Sprint: A development iteration lasting 2-4 weeks. Sprint 0 is a planning sprint.**

**5.KP:** A model is a simplified, abstract view of a system, that can show context, interactions, structure and behavior.  
**Context models show a system relates to other systems and processes.**Use case and sequence diagrams describe how a user interacts with the system. Use case show interactions between a system and external actors; sequence shows interactions between system objects.  
**Structural Models show the organization and architecture of the system. Class diagrams are used to define the static structure of classes in a system.**Behavioral models are used to show the dynamic behavior of the system, showing data processed or events that cause responses from the system.   
**Activity diagrams model the processing of data.**State diagrams are used to model responses of the system to internal or external events.  
**Model driven engineering involves representing a system as a set of models that can be transformed to executable code automatically.**UML: Unified Modeling Language

**C5LD:** Requirements are everything that must be true about the SW for it to be acceptable and nothing more. What, but not How**.**    
**Kano Model shows how Delighters, Performance, and Basic needs result in satisfaction.**

**3.Lec:** Silver Bullet Problems getting error-free SW systems: **Essential:** Complexity, Conformity, Invisibility, Changeability. **Accidental:** Human error, Bad interfaces, inadequate abstractions, lack of math/engineering foundation. \*\*\* **Life Cycle:** Requirements Analysis and Specification, **Architecture and Design,** Implementation, **Testing/QA and Validation,** Deployment and Maintenance, **Evolution(Repeat).** \*\*\* Requirements are acquired from the customer, and tell what the SW is supposed to do. **Specification is derived from the requirements, and tell how the system will accomplish that task.** \*\*\* Perspective Models: **Planned,** Ordered approach, defined structure, **Helps control decision making and product management.** Agile Models: **Simple,** Adaptive

**4.KP:** Requirements set out what a system should do and defines constraints. **Functional requirements are statements of the services that the system must provide or how computations must be carried out.** Non-Functional requirements often constrain the system and processes used. These might be product, organizational, or external requirements. **Requirements Engineering includes elicitation, specification, validation, and management.** Elicitation involves requirement discovery, classification and organization, negotiation, and documentation. **Specification is formally documenting user and system requirements, and generating a SW requirements document.** A SW requirements document is a list of system requirements, accessible to both the developer and the customer. **Validation is checking the requirements for validity, consistency, completeness, realism, and verifiability.** Management is handling any changes to the requirements.

**3.KP:** Agile Methods are iterative development methods that focus on reducing overheads and documentation, and incremental SW delivery. Involves customers directly in the development process. **Plan driven or Agile methods should be chosen based on the team working and the SW being developed. Most often, a mix is used.** Scrum is an agile method that offers a framework for organization. It involves sprints, which are fixed times when an increment is developed. Planning involves prioritizing tasks for each sprint.

**2.Lec:** Focus of SE: **Organizing teams to cooperatively build systems**, determining what to build, **architecture**, analysis and testing, **and lifecycle system engineering**. \*\*\* Ethical Principles: **Public,** Client and Employer, **Product,** Judgement and Integrity, **Management,** Profession, **Colleagues,** self. \*\*\* **Contracts:** Non-Disclosure Agreement, **Non-Compete Agreement**, Work for Hire/Vendor /Contractor.

**1.kp:** SE is an engineering discipline that is concerned with all aspects of software production. **Software includes all documentation, QA, and developers.** Software attributes are Maintainability, dependability, efficiency, and acceptability. **The software process includes specification, development, validation, and evolution.** Few, if any, specific design and implementation techniques are applicable to all kinds of system. **SE fundamentals are managing software processes, dependability and security, requirements engineering, and reuse.** Professional societies publish codes of conduct that cover ethical and professional standards.