Lecture 1:**SOFTWARE LIFE CYCLE MODELS** Some popular software life cycle models: 1. Waterfall 2. Build-and-Fix 3. Rapid Prototype 4. Spiral 5. Incremental 6. V-Model 7. Synchronize-and-Stabilize 8. Evolutionary/Technology-Based 9. Evolutionary 10. Rapid Application Development (RAD) 11. Concurrent Development 12. Formal Methods 13. 4th Generation 14. Object-Oriented Models Note: a model is a representation of a process or system **Waterfall** (or Linear Sequential or Classic) Model Phases follow one another in linear order, with iteration. **Build and Fix** build first version, keep modifying until client is satisfied **V Model** SQA team performs verification & validation (V&V) after each phase of life cycle, culminating in final overall V&V **SOFTWARE LIFE CYCLE PHASES** Explicit Software Life Cycle Phases(shown in life cycle models)1. Conception 2. Requirements Analysis 3. Design 4. Implementation (Programming) 5. Testing/Verification/ Validation 6. Maintenance 7. Retirement Some Implicit Software Life Cycle Phases (not shown in life cycle models) 1. Management 2. Documentation 3. SQA **Some Major Reports (Documentation) for Life Cycle Phases:** 1. Software Requirements Specifications (SRS) 2. Project Management Plan (PMP) 3. Software Quality Assurance Plan (SQAP) 4. Design Document (DD) 5. Test Plan (TP) 6. Final Report (all of the above) Some Diagrams Used for Documenting the Phases: 1. UML Diagrams (Use Case, Sequence, Statechart, . . . ) 2. Entity-Relationship (ER) Diagrams 3. Gantt Chart 4. and so on , . .

**EXERCISE 1a**: Consider the list of implicit life cycle phases already mentioned. Note that implicit phases do not show up in most life cycle models. Keeping this in mind, give one or two additional implicit life cycle phases: **ANSWER** Project Management Plan (PMP), Software Verification and Validation Plan (SVVP), Risk Management Monitoring & Mitigation Plan (RMMM), Software Quality Assurance Plan (SQAP), **EXERCISE 1b True:** Most implicit life cycle phases tend to go on for the entire life cycle.

THE CONCEPTION PHASE 1. Conception Input: <various> Action: concept of a product Output Documentation: vague statement of product

THE REQUIREMENTS PHASE 2. Requirements Analysis Input: vague statement of product Action:

System analyst (or software engineer) interacts with customer to understand the needs (requirements) of customer Output: software requirements specification (SRS), precise, detailed, and unambiguous statement of what will be delivered to customer (what software does)

NOTE: A distinction is sometimes made between requirements and specification Requirements = what customer wants Specifications = what is delivered to customer ideally, requirements should equal specifications

THE DESIGN PHASE 3. Software Design Input: software requirements specification (SRS) Action:

Software designer develops architectural and detailed description of software Output: design document (DD), technical description of software, from overview to specifics (how software works)

IMPLEMENTATION PHASE 4. Implementation (previously programming) Input: design document (DD) Action: programmer generates code using tools and/or programming languages Output: test plan (TP), program modules (code)

TESTING PHASE 5. Testing (or SQA/Test/V&V) Input: test plan (TP) and program modules Action:

Program modules are integrated and tested Output: working product and maintenance/operation documentation

OPERATION/MAINTENANCE PHASE 6. Operation and Maintenance Input: working product and maintenance/operation documentation Action: product is used, defects are fixed, and changes/upgrades are made Output: evolving product and maintenance record

RETIREMENT PHASE 7. Retirement Input: evolving product Action: product is discarded and/or replaced (cycle begins again) Output:

**Lecture 2:** A metric is a numeric measurement. A software metric is a numeric software measurement. Three popular software metrics, previously discussed, are: PM = Person-Month (previously called Man-Month) LOC = Lines of Code (1000 LOC = 1 KLOC) FP = Function Points These three metrics are sometimes referred to as the primary metric variables. A fundamental primary metric is time, but will not be our focus since it is not a basic software metric, and it is included in Person-Month. Several other metrics, called secondary or computed metrics, are based on the primary metrics. Note: Hewlett-Packard defines and uses six primary metrics.. LOC = LINES OF CODE: Measured by 1.Each Statement, Comments, statement delimiters



































