Process Model: Specialized Process Models

LECTURE # 11

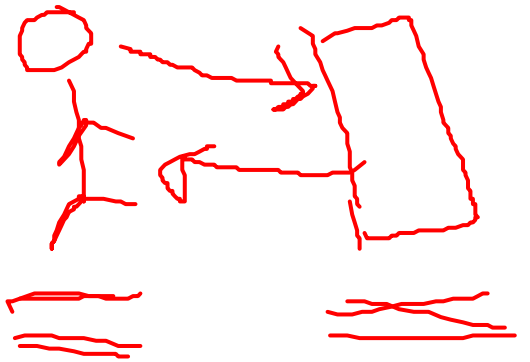
Lecture by Engr. Sidra

Unified Software Process Model

- The Unified Software Development Process is an industry standard software engineering process
- It is commonly referred to as the "Unified Process" or UP
- The Unified Process is component based
- The Unified Process uses the Unified Modelling Language for documentation and design
- UP is a generic software engineering process. It has to be customized (instantiated) for your project
  - In house standards, document templates, tools, databases, lifecycle modifications, …
- UP is:
  - Use case (requirements) driven
  - Architecture centric
  - Iterative and incremental
  - Risk driven

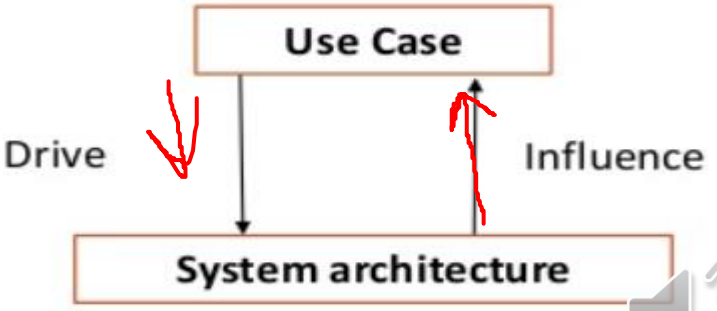
Use-Case Driven

- **Use-Case Driven means**:
  - Development process proceeds through a series of workflows that derive from use cases.
  - Tool for specifying requirements
  - Driving design
  - Source for testing
- **Terminologies**
  - **Users:** Someone or something that interact with systems
  - **Use Case**: interaction between users and system, what the system supposed to do for each user?
  - **Use Case Model**: collection of users; description of complete functionality



Architecture-Centric

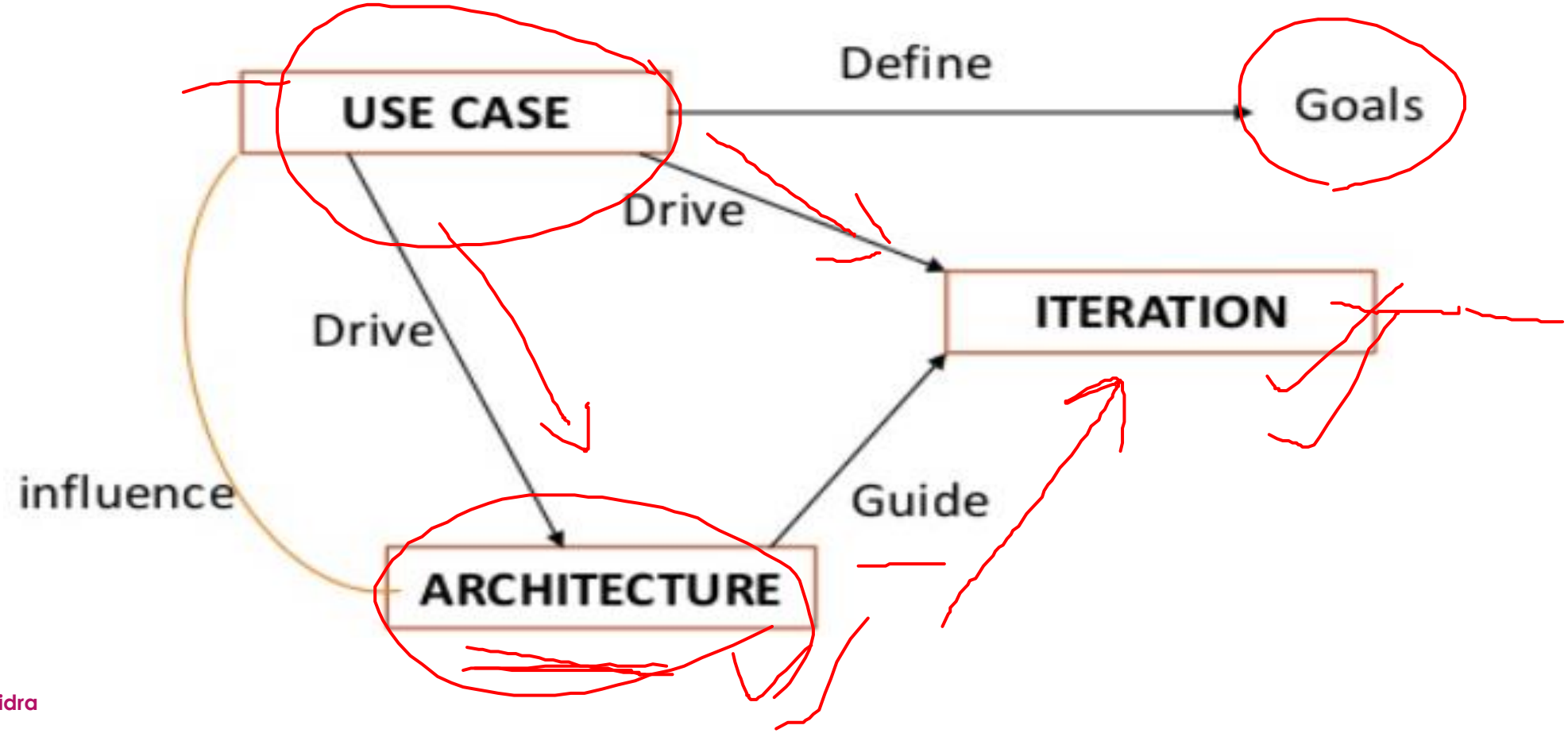
- Architecture is the view of the whole design with key Characteristics and without too many details
- Growth with use case in parallel (structure and function)
- Simplified Process
  - Rough outline (use case independent)
  - Subset of identified use cases (5-10%)
  - More use cases specified, more architecture discovered



Iterative and Incremental

- **Iteration:** Steps in the workflow (mini-project)
  - Create a design for relevant use cases
  - Implement with components
  - Required iteration in logical order for economy
- **Incremental:** Growth in the product (might not be additive)

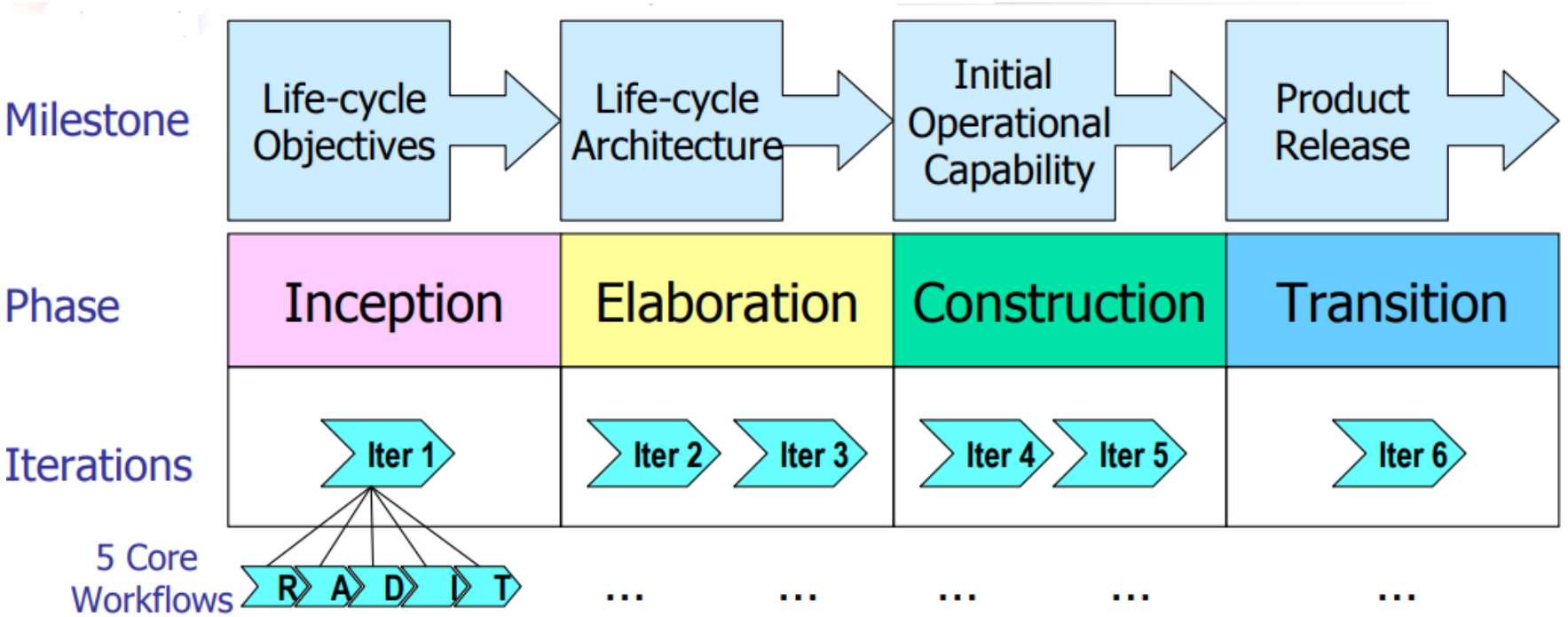
Relationship of 3 Concepts



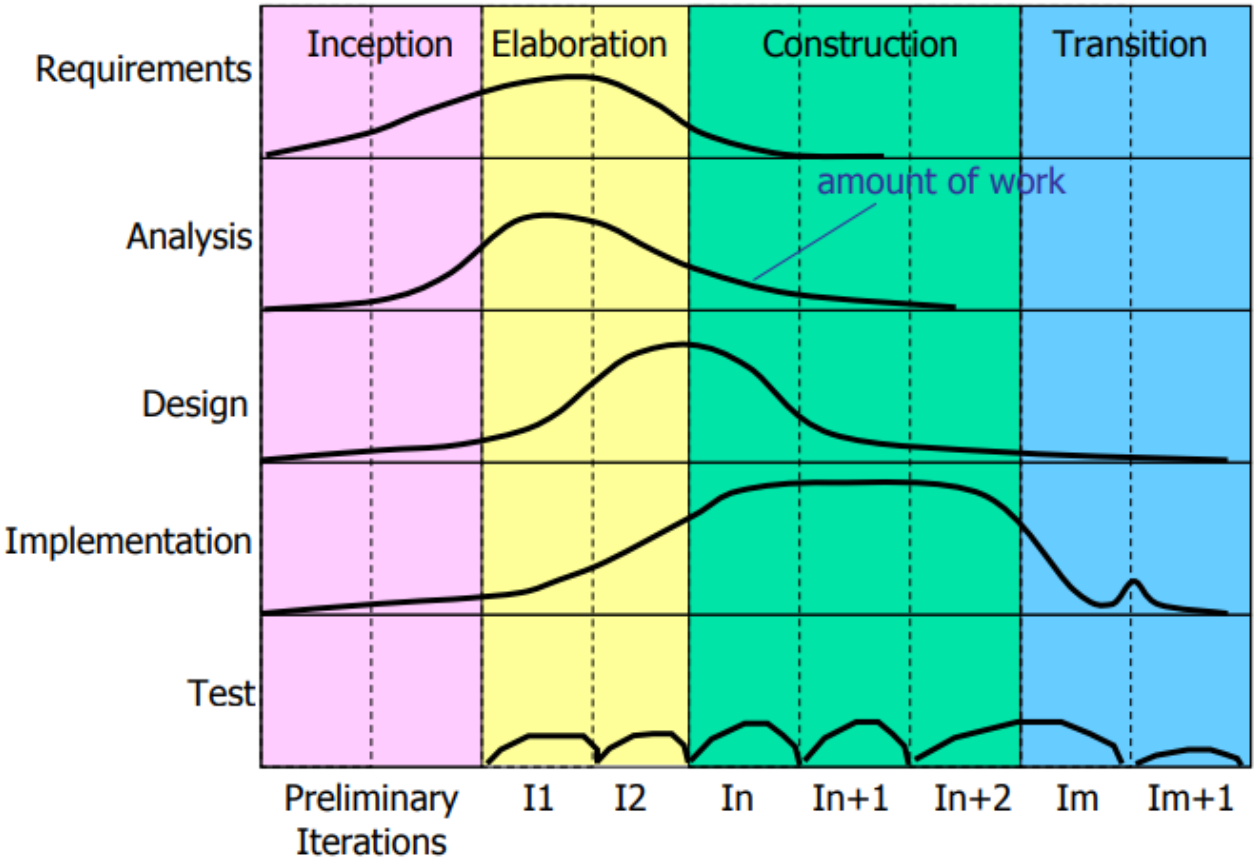
Risk Driven

- The Risk-Driven Model helps developers decide how much architecture work to do.
- The essence of the Risk-Driven Model is these three steps:
  - 1) Identify and prioritize risks
  - 2) Select and apply a set of architecture techniques
  - 3) Evaluate risk reduction
- It helps to prioritize your risks, apply chosen techniques, then evaluate any remaining risk, which means that you must decide if the risk has been sufficiently mitigated.

UP Structure



Phases and Workflow



Advantages of UP

- Early risk management and mitigation
- Robust architecture at early stage
- Change is more manageable
- Better training of team in workflows
- Higher level of reuse
- Higher overall product quality
- Less time is required for integration as the process of integration goes on throughout the software development life cycle

Disadvantages of UP

- The team members need to be expert in their field to develop a software under this methodology.
- The development process is too complex and disorganized.
- On cutting edge projects which utilize new technology, the reuse of components will not be possible. Hence the time saving one could have made will be impossible to fulfill.
- Integration throughout the process of software development, in theory sounds a good thing. But on particularly big projects with multiple development streams it will only add to the confusion and cause more issues during the stages of testing
- Heavy documentation can be expensive