Software Engineering

Robert A. Conchada

DICT-III

Lesson 3

1. **Define Activity Framework.**

Defining a Framework Activity

A software team would need significantly more information before it could properly execute any one of these activities as part of the software process. Therefore, you are faced with key questions: What actions are appropriate for a framework activity, given the nature of the problem to be solved, the characteristics of the people doing the work, and the stakeholders who are sponsoring the project?

For a small software project requested by one person (at a remote location) with simple, straightforward requirements, the communication activity might encompass little more that a phone call with the appropriate stakeholder. Therefore, the only necessary action is phone conversation, and the work tasks (the tasks set) that this action encompasses are:
1. Make contact with stakeholder via telephone.
2. Discuss requirements and take notes.
3. Organized notes into a brief written statement of requirements.
4. E-mail to stakeholder for review and approval.

If the project was considerably more complex with many stakeholders, each with a different set of requirements, the communication activity might have six distinct actions: inception, elicitation, elaboration, negotiation, specification and validation. Each of these software engineering actions would have many work tasks and a number of distinct work products.

2. **Enumerate the prescriptive Process Models**

Prescriptive process models were originally proposed to bring order to the chaos of software development. History has indicated that these traditional models have brough a certain amount of useful structure to software engineering works and have provided a reasonably effective road map for software teams. However, software engineering work and the product that it produces remain on "the edge of chaos".

3. Provide examples of software projects that would be amenable to the incremental model.

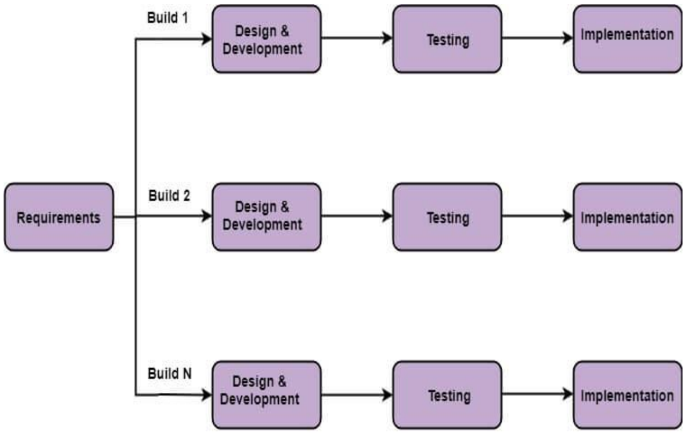

Fig: Incremental Model

Incremental development is particularly useful when staffing is unavailable for a complete implementation by the business deadline that has been established for the project. Early increments can be implemented with fewer people. If the core product is well received, then additional staff (if required) can be added to implement the next increment. In addition, increments can be planned to manage technical risks. For example, a major system might require the availability of new hardware that is under development and whose delivery date is uncertain.It might be possible to plan early increments in a way that avoids the use of this hardware,thereby enabling partial functionality to be delivered to end users without inordinate delay.

4. Develop a set of actions for the communication activity.

Prototyping. Often, a customer defines a set of general objectives for software but does not identify detailed requirements for functions and features. In other cases, the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human-machine interaction should take. In these, and many other situations, a prototyping paradigm may offer the best approach.

By communication, customer requirement gathering is done. Communication with consumers and stakeholders to determine the system's objectives and the software's requirements.