Software Project Management 4th Edition

Chapter 3

Project evaluation & Estimation

Main topics to be covered

- Software Life Cycle models.

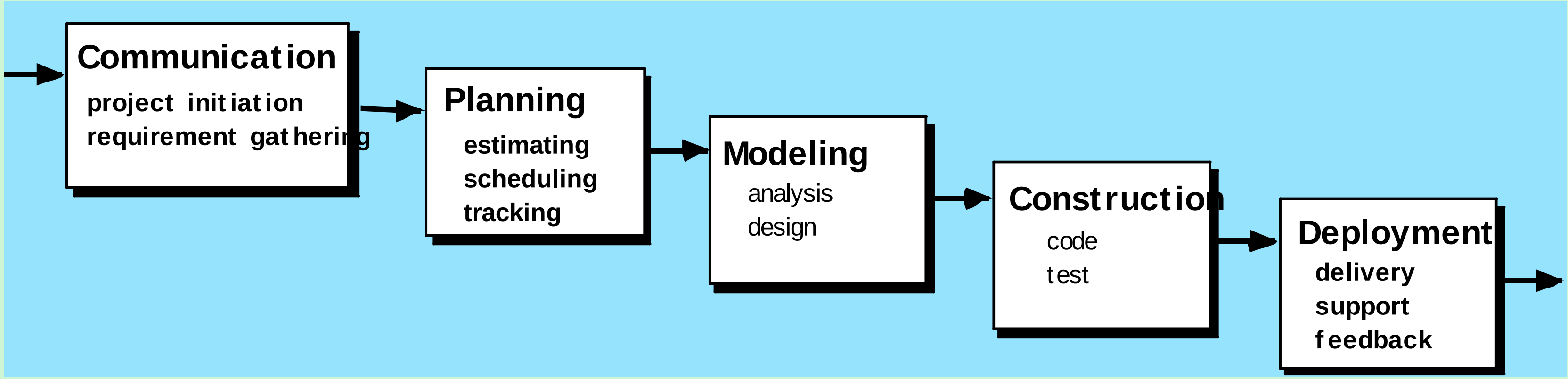
Software Life Cycle Models

- “The period of time that starts when a software product is conceived and ends when the product is no longer available for use.
- The software life cycle typically includes a requirement phase, design phase, implementation phase, test phase, operation and maintenance phase”.

The Waterfall Model

- The waterfall model sometimes called the classic life cycle, suggests a systematic, sequential approach to software development that begins with customer specification of requirements and progresses through planning, modeling, construction, and deployment, culminating in on-going support for the complete software.
- The waterfall model is the oldest paradigm(model) for software engineering.

The Waterfall Model



Waterfall Model

This model is easy to understand and reinforces the notion of “define before design” and “design before code”.

The model expects complete & accurate requirements early in the process, which is unrealistic

Waterfall Model

Problems of waterfall model

i. It is difficult to define all requirements at the beginning of a project

ii. This model is not suitable for accommodating any change

iii. A working version of the system is not seen until late in the project's life

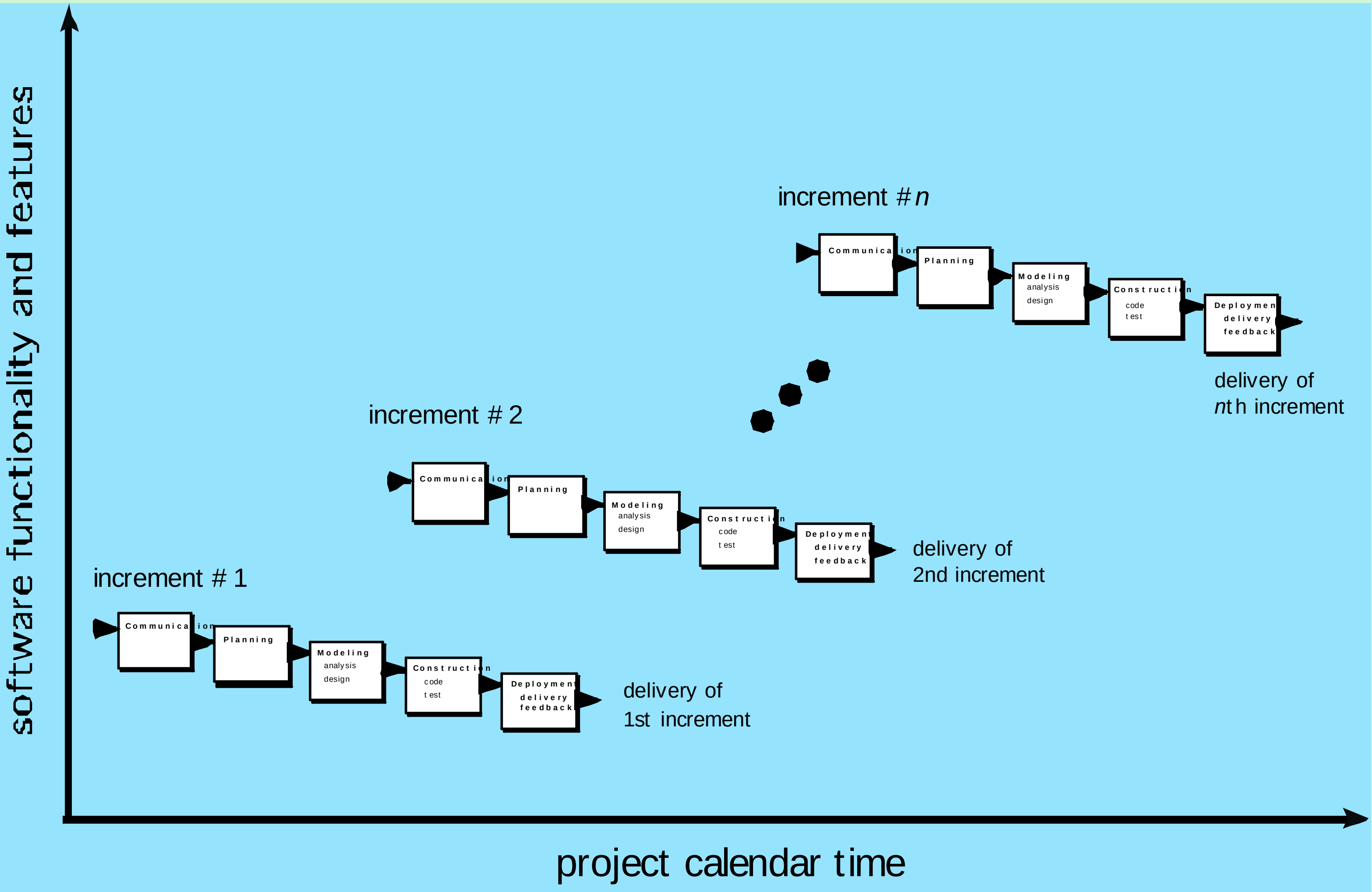
iv. It does not scale up well to large projects.

v. Real projects are rarely sequential.

Incremental process models

- The incremental model
- The RAD model

The Incremental Model



The Incremental Model

- The incremental model applies linear sequences in a staggered(spread over a period of time) fashion as calendar time progresses.
- Each linear sequence produces deliverable “increments” of the software. For eg. Word processing software developed using the incremental paradigm might deliver basic file management editing, and document production functions in the first increment;
- More sophisticated editing, and document production capabilities in the second increment; spelling grammar checking in the third increment; and advanced page layout capability in the fourth increment.

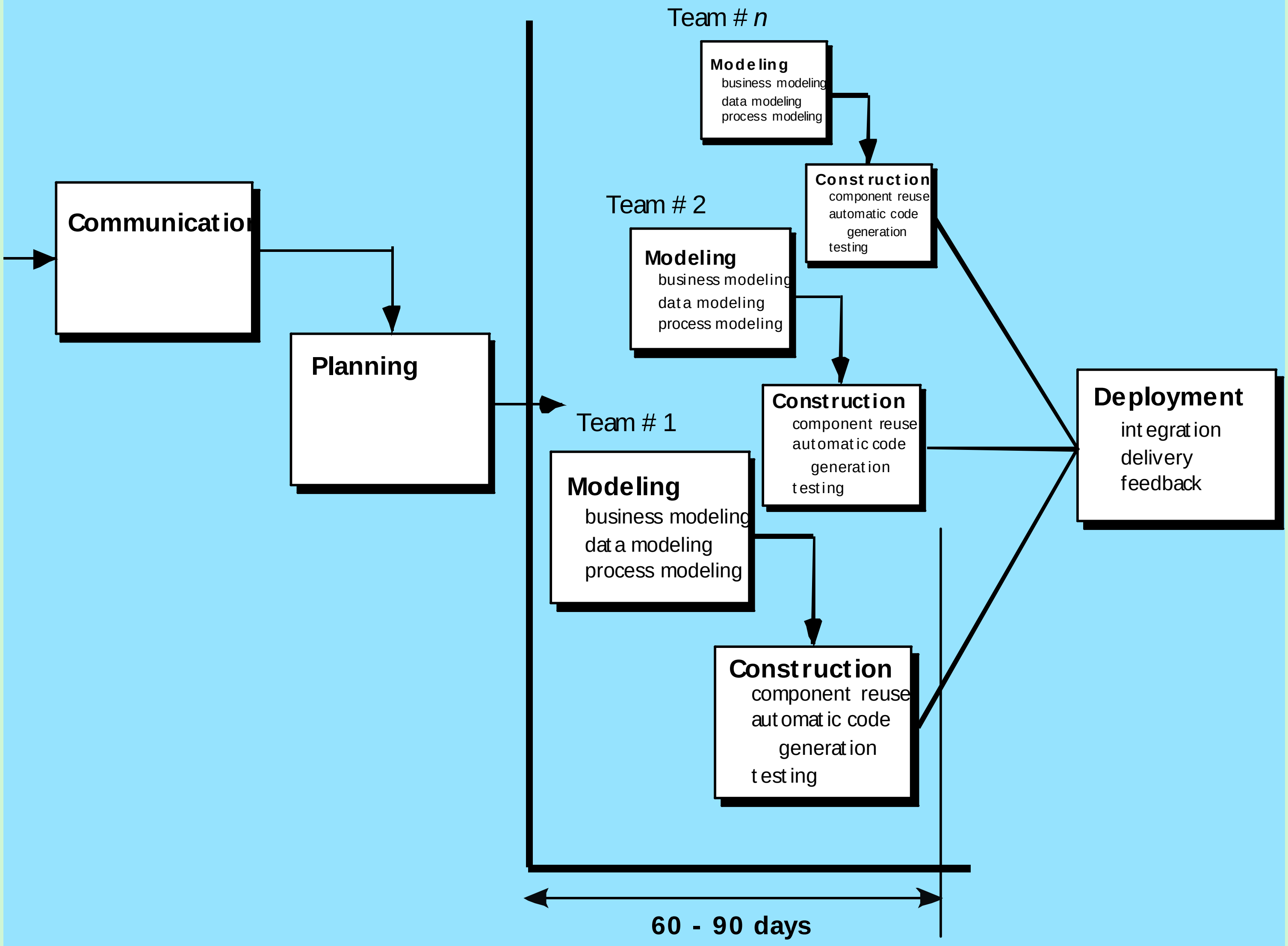
- When the incremental model is used, the first increment is often a core product; i.e. basic requirements are addressed, but many supplementary features known, others unknown) remain undelivered.
- The core product is used by the customer. As a result of use and/or evaluation, a plan is developed for the next increment.
- The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality. This process is repeated following the delivery of each increment, until the complete product is produced.

Uses

- This model 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, additional staff(if reqd) can be added to implement the next increment.

- In addition, increments can be planned to manage technical risks. For eg. 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, there by enabling partial functionality to be delivered to end users without unreasonable delay.

The RAD Model



The RAD Model

- ***Rapid Application Model(RAD)*** is an incremental software process model that emphasizes a short development cycle.
- The RAD model is a" high-speed" adaptation of the waterfall model, in which rapid development is achieved by using a component based construction approach.
- If requirements are well understood and project scope is constrained (controlled) the RAD process enables a development team to create a "fully-functional system" with in a very short time period (eg. 60 to 90 days).

- ***Communication*** works to understand the business problem and the information characteristics that the software must accommodate.
- ***Planning*** is essential because multiple software teams work in parallel on different system functions.
- ***Modeling*** encompasses three major phases – business modeling, data modeling, process modeling – and establishes design representations that serve as the basis for RAD's construction activity.
- ***Construction*** emphasizes the use of pre-existing software components and the application of automated code generation.
- Finally, ***deployment*** establishes a basis for subsequent iterations, if required.
- The RAD process model is illustrated in fig. Obviously the time constraints imposed on a RAD project demand a "scalable scope"

- If a business application can be modularized in a way that enables each major function to be completed in less than three months (using the above described approach), it is a candidate for RAD. Each major function can be addressed by a separate RAD team and then integrated to form a whole.
- Like all process models, the RAD approach has drawbacks:

(1) For large, but scalable projects, RAD requires sufficient human resources to create the right number of RAD teams.

(2) If developers and customers are not committed to rapid fire activities necessary to complete the system in a much abbreviated (reduced) time frame, RAD projects will fail.

(3) RAD may not be appropriate when technical risks are high (eg. When a new application makes heavy use of a new technology)

Applications

- If a business application can be modularized in a way that enables each major function to be completed in less than three months (Life cycle models and approaches like candidate for RAD Model). Each major function can be addressed by a separate RAD team and then integrated to form a whole.

Research

Steps in the Research Life Cycle

1. Proposal Planning & Writing
2. Project Start Up
3. Data Collection
4. Data Analysis
5. Data Sharing
6. End of Project

Reference Link:

http://www2.lib.virginia.edu/brown/data/lifecycle.html