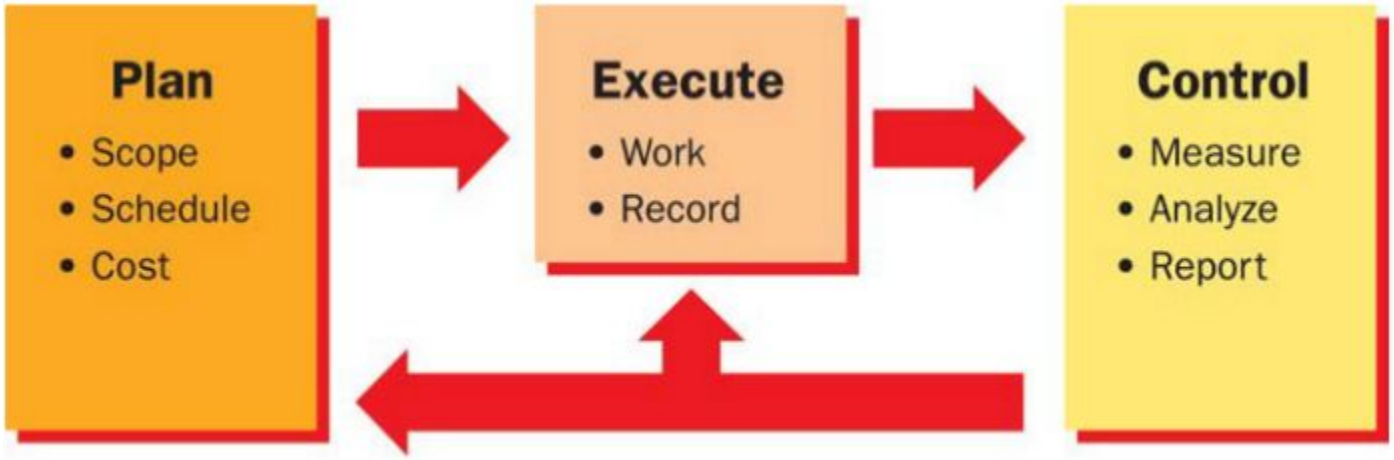
Métodos de Desenvolvimento de Software (MDS)
2016/2017
Lecture 2

Vasco Amaral

Project Management

Managing projects

To manage a given project means to Plan, Execute and control the Work Activity of that same project



Plan

To Plan a given project means to understand:

- **What** is the work that must be acomplished(scope) and what are their correponding activity components (Work Breakdown Structure);
- **Who** is going to execute and manage the work to be done (responsability matrix)
- **When** is the work going to be done (calendar)
- **Cost** of work, materials and other required resources for its acomplishment

Execute

To execute a project means:

- to **accomplish** the work that needs to be done according to what was planned;
- and **keep informed** the team and managers.

Control

- Monitor and Report the execution of the management plan pertaining to: scope, time and cost, as well as quality and risk.

scope

cost

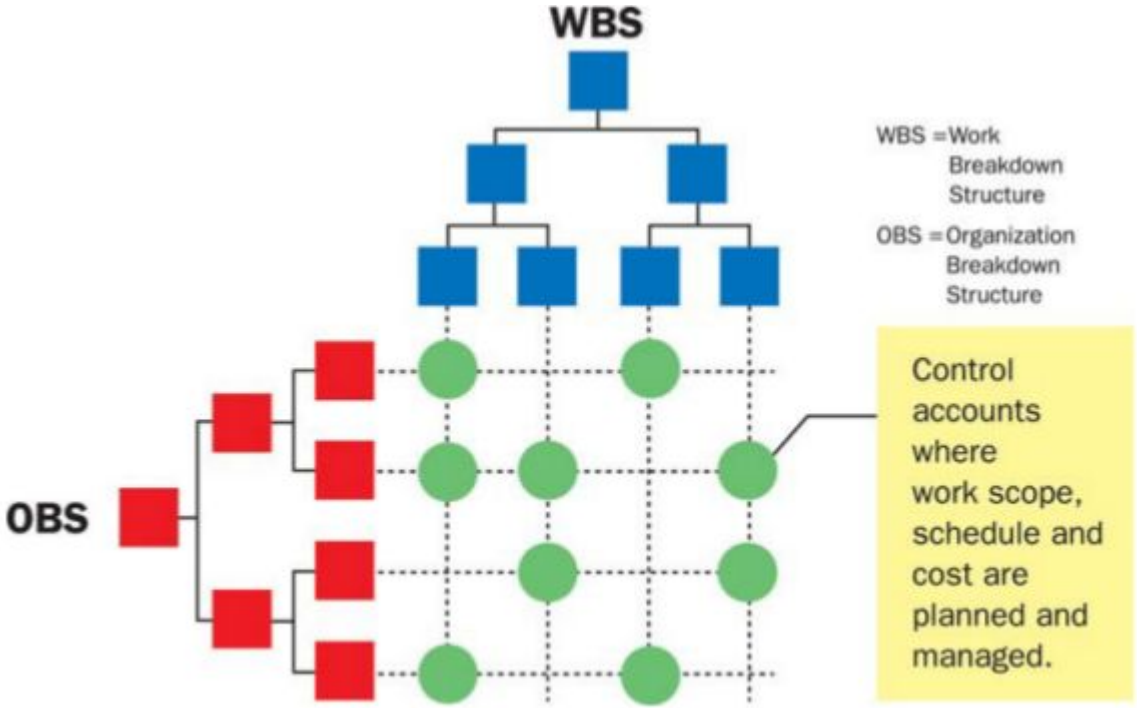
Time

Major goal: to keep work performance and its results aligned with the initial plans, within a tolerance margin.

Software project planning

- Identify activities
- Schedule activities
- Assign resources

Activities and resources: scheduling?



- **Work Breakdown Structure (WBS)** – Decomposes the work to be done in a set of activities.
- **Organization Breakdown Structure** – Create the structure of the organization and is useful for relating elements to the project activity.

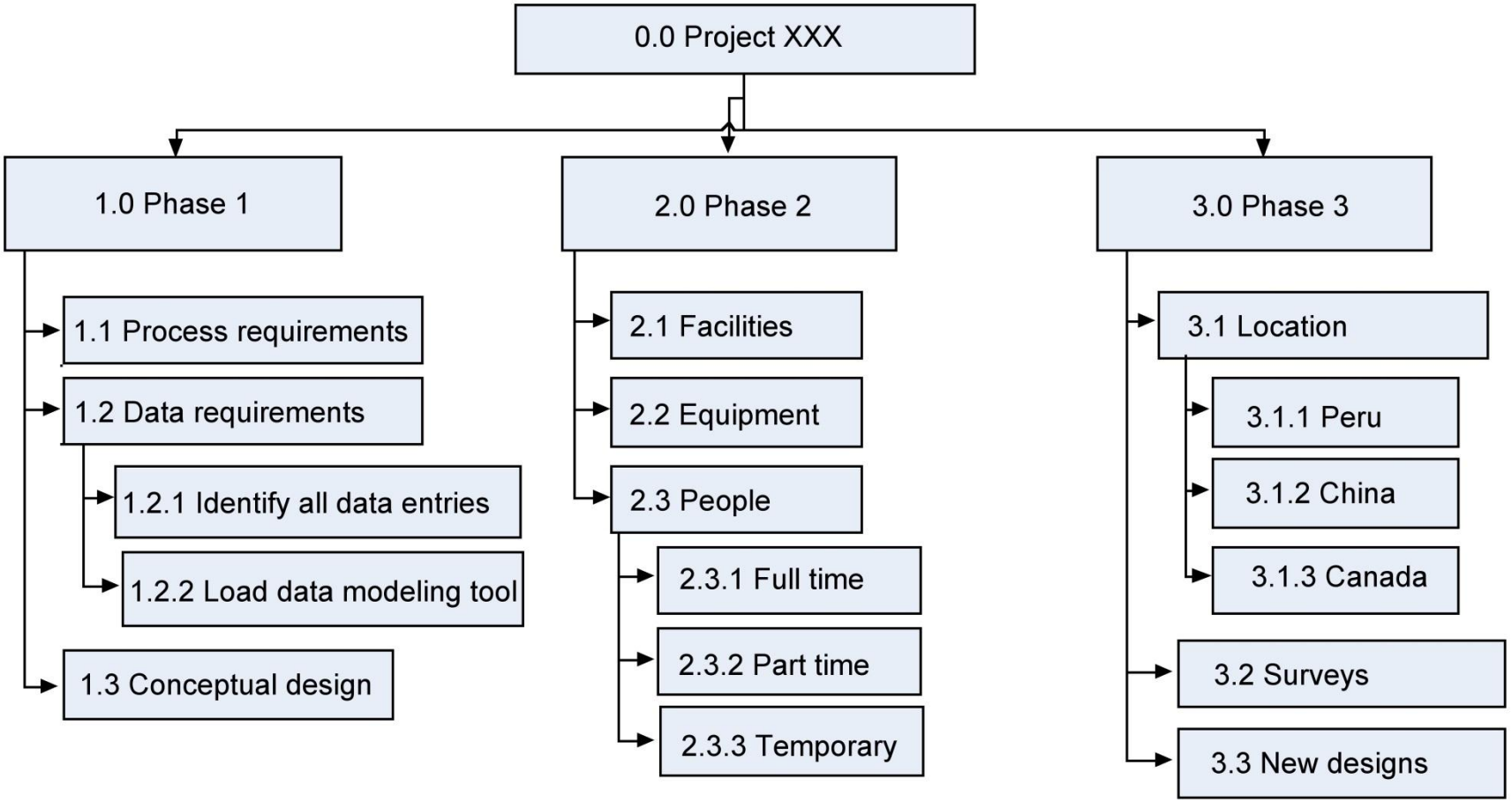
Activity identification

Work Breakdown Structure

Work Breakdown Structure (WBS)

- Defines the scope of the project ("to do" list)
- Breaks work down to components
  - Subdivides complex tasks into simpler ones
- Hints on building a WBS
  - Include 100% of the work / deliverables
  - Avoid overlapping elements
  - Plan first for work outcomes, rather than actions
  - Try to get the "sweet spot" of detail
    - Too little makes planning harder
    - Too much hinders the communication role of the WBS

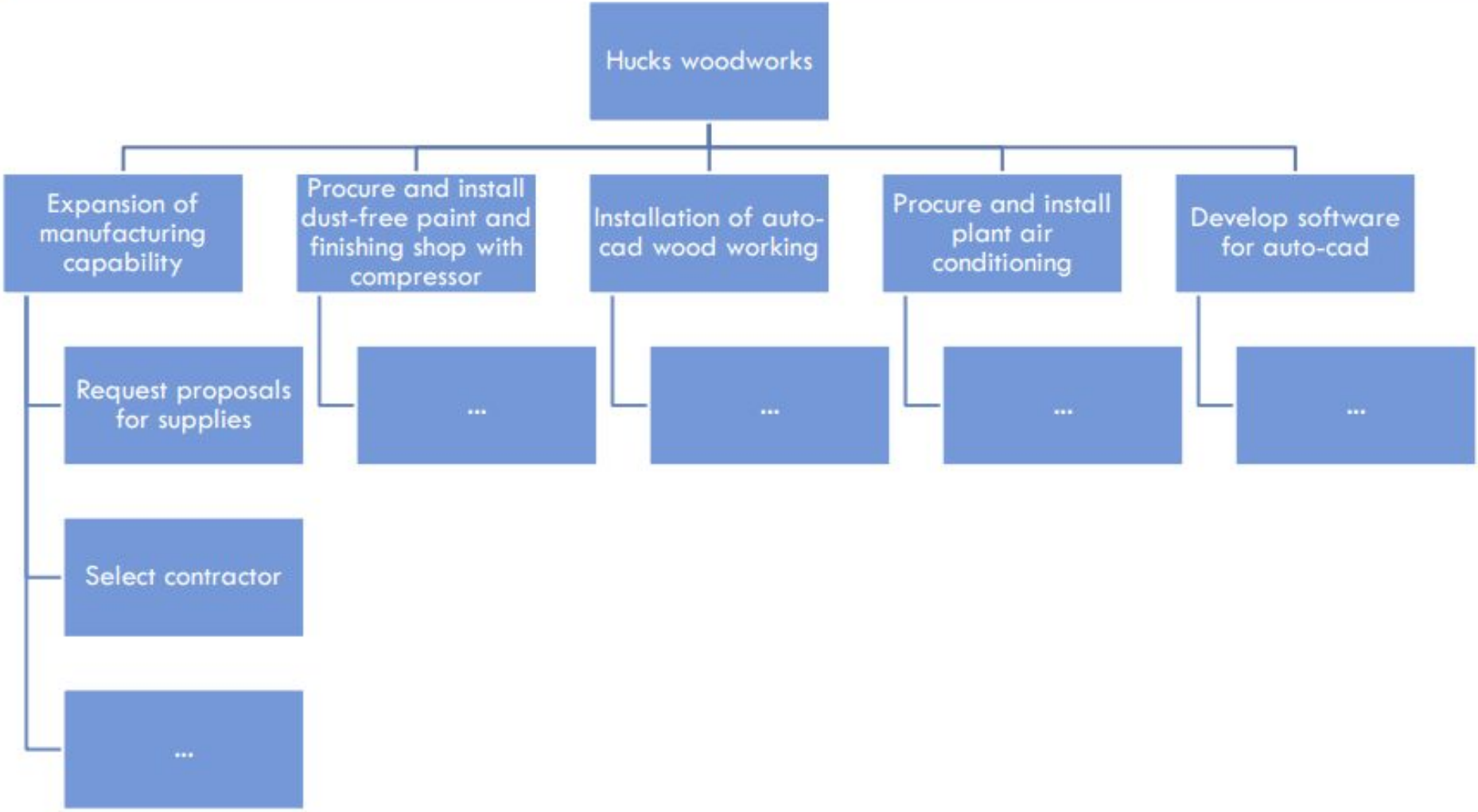
WBS - Example


0.0 Project XXX
1.0 Phase 1
2.0 Phase 2
3.0 Phase 3
1.1 Process requirements
1.2 Data requirements
1.2.1 Identify all data entries
1.2.2 Load data modeling tool
1.3 Conceptual design
2.1 Facilities
2.2 Equipment
2.3 People
2.3.1 Full time
2.3.2 Part time
2.3.3 Temporary
3.1 Location
3.1.1 Peru
3.1.2 China
3.1.3 Canada
3.2 Surveys
3.3 New designs

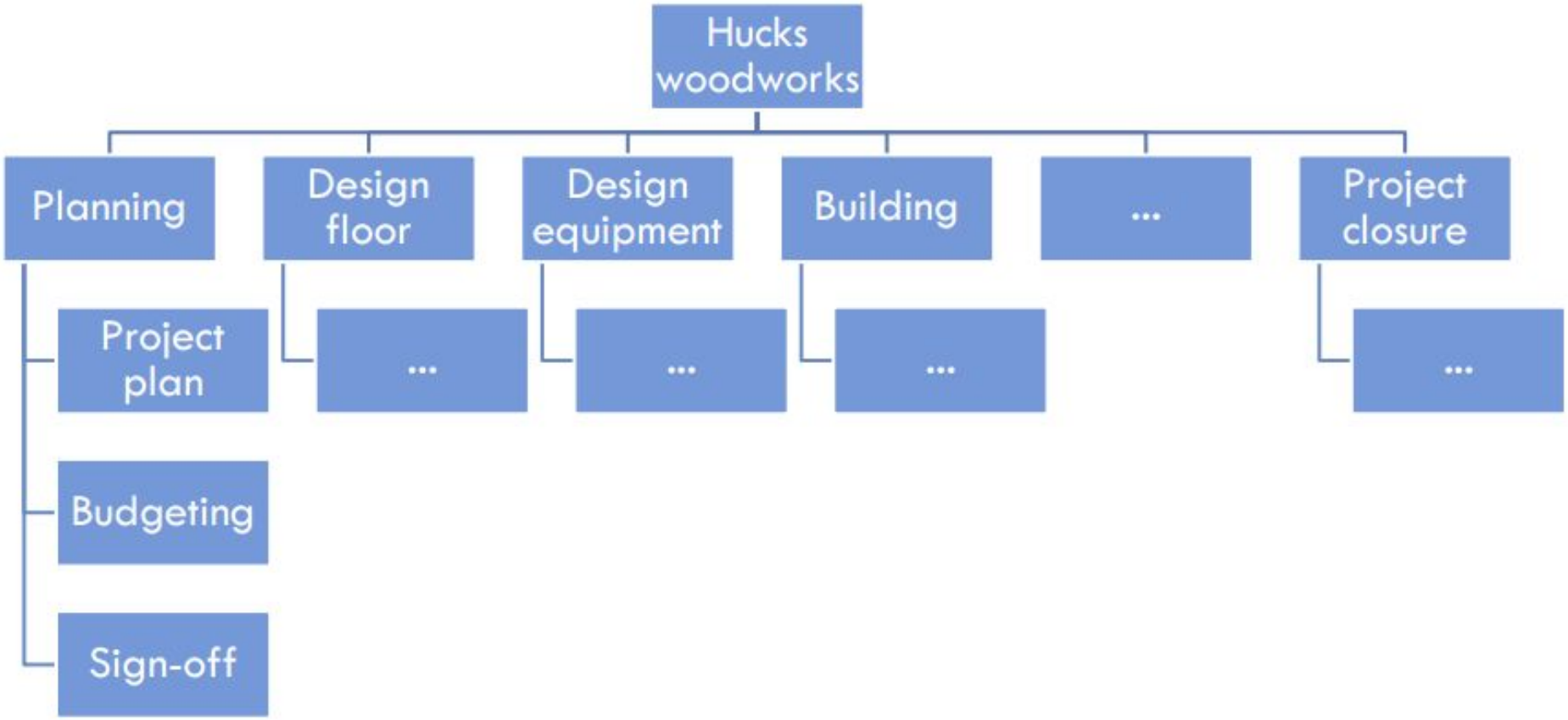
Creating WBSs

- Several possible decomposing strategies
  - Product oriented
  - Process oriented
  - ...
- The WBS may have a different number of levels
  - Enough to facilitate estimates on costs, resources, ...
  - Not too many, to facilitate communication
- The lower level components are **work packages**
  - Must be assigned to individuals, or teams, responsible for delivering them
  - Estimates of time, costs and resources are done at the work package level of granularity

Hucks Woodworks (product oriented)



Hucks Woodworks (process oriented)



How much detail (finding the sweet spot)

- The 8/80 rule (of thumb)
  - No work package should be less than 8 hours or more than 80 hours
  - Groups of tasks, or activities, once complete, should correspond to the completeness of the corresponding upper level

Scheduling activities

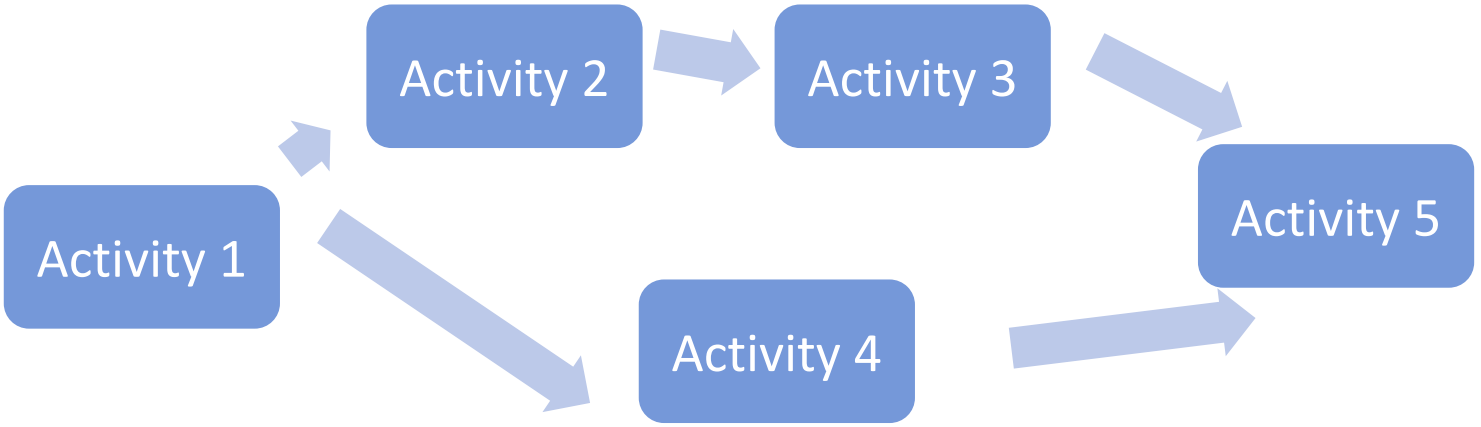
Naïve project scheduling

- Project:
  - A sequence of interconnected activities to achieve a certain goal

- Problem:
  - Longest possible completion schedule

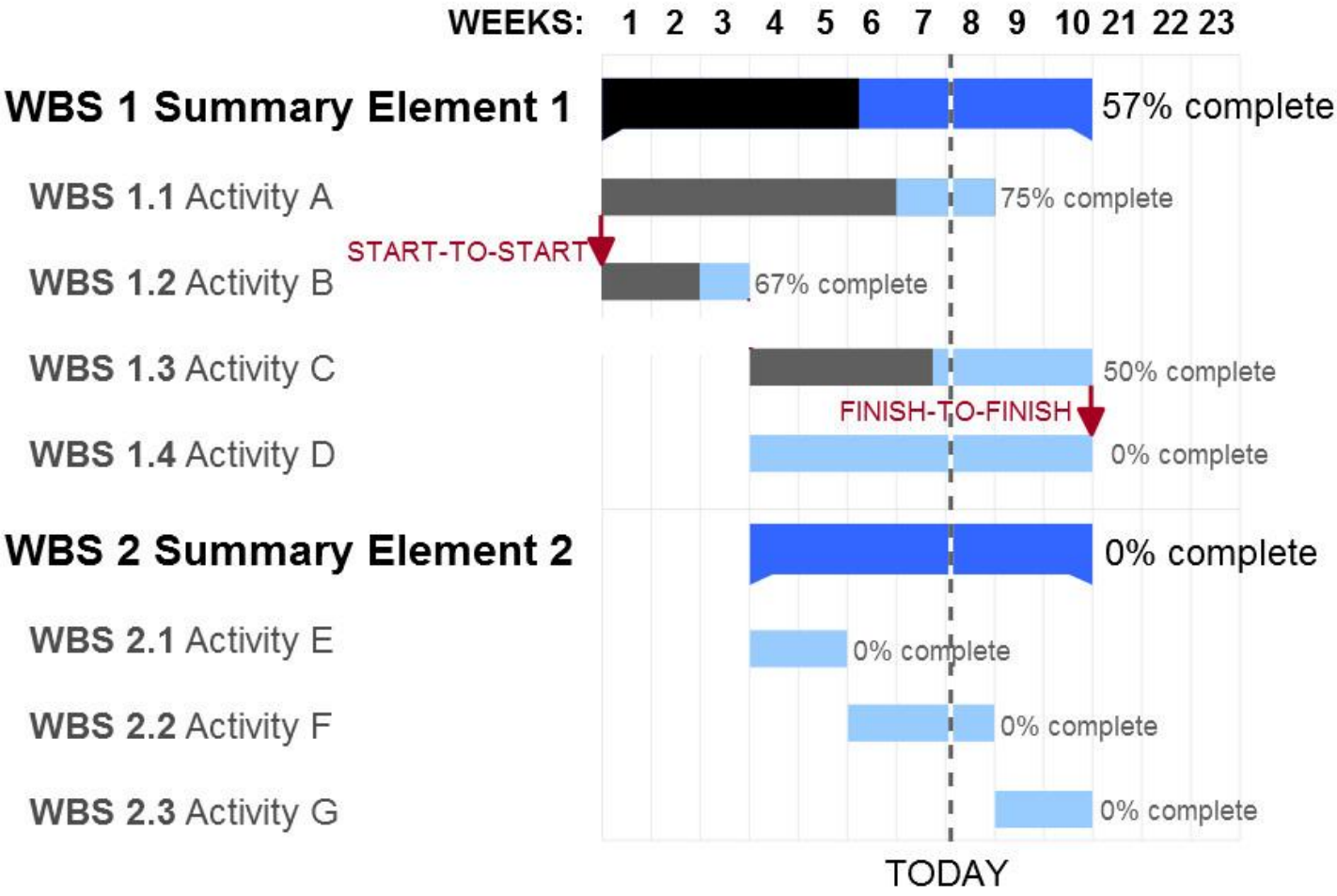
Networked project plan

- Build a network of relationships among activities, so that activities precedences can be established
- Which activities must finish, before a particular new activity starts?



Gantt charts

Gantt Charts (illustration)



Gantt charts limitations

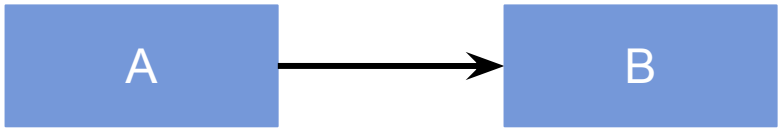
- Lack of detail on task precedence
- No support for helping the project manager
- Defining the shortest possible completion schedule
- Allocating resources effectively

Some comercial tools make use of dependence relations, like the ones described next...

Dependence relationships (FS)

Finish to **S**tart relationship

- As soon as A finishes, B can start



Example

- A is a data collection activity
- B is a data storing activity
- As soon as we finish data collection, we can start data storing

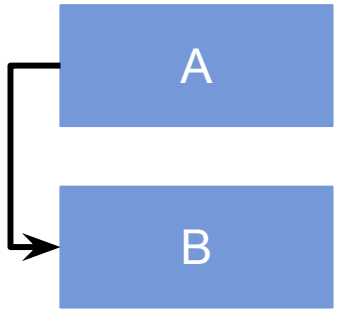
Notes:

□Recommended as default dependency in early planning

Dependence relationships (SS)

Start to **S**tart relationship

- As soon as A starts, B can start



Example

- A is a data collection activity
- B is a data storing activity
- Data storing cannot start before data collection starts

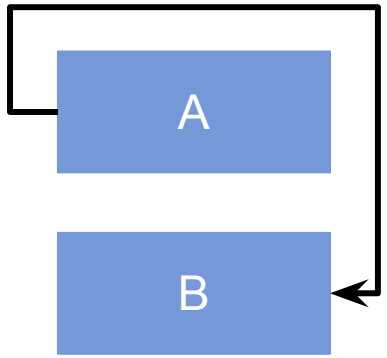
Notes:

- Use for compressing activities

Dependence relationships (SF)

Start to **F**inish relationship

- As soon as A starts, B can finish



Example

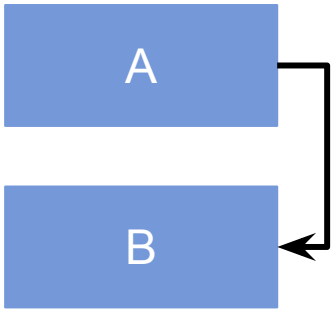
- A is a new system running
- B is an old system running
- As soon as the new system is running, the old system may be discontinued

Notes:

- Use for just in time scheduling (relatively uncommon)

Dependence relationships (FF)

Finish to **F**inish relationship



- As soon as A finishes, B can finish

Example

- A is a data collection activity
- B is a data storing activity
- Data storing cannot finish before data collection finishes

Notes:

- To preserve connectivity in the network, **SS** should be accompanied with **FF**

Activity On Arrow diagrams

(the most common of these is the Pert diagram)

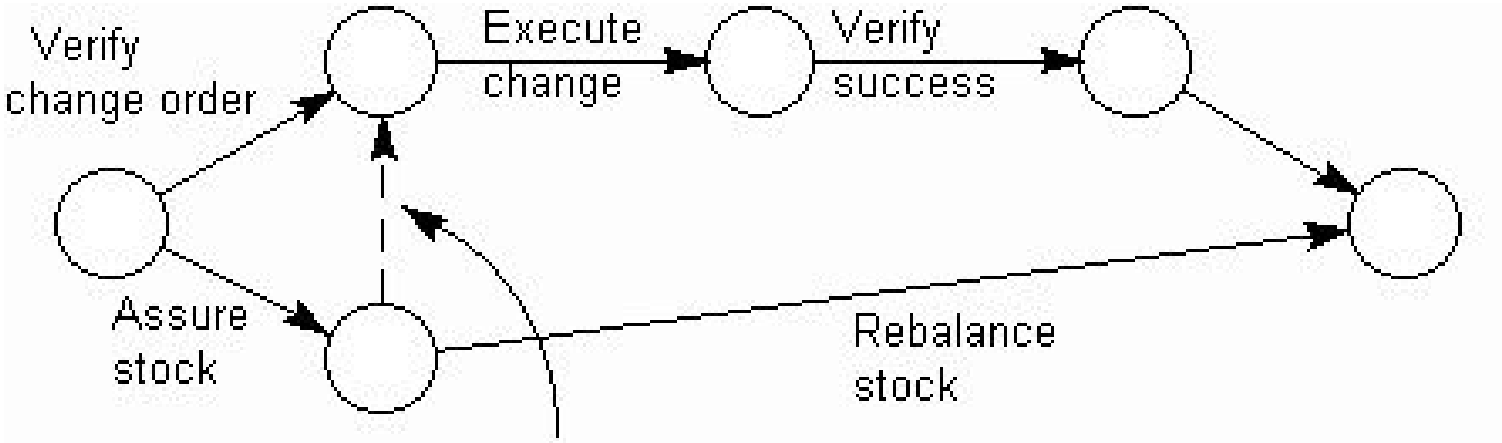
Pert diagrams

- Detailed project planning & control
- Support for project schedule analytics
  - Identification of the first possible moment for completing an activity
  - Support for earliest completion date computation
  - Comparison of alternative detailed scheduling
  - Project scheduling control

Activity On Arrow (AOA) Diagrams

- Arrows represent activities execution
- Nodes represent the start (or end) of activities
- Dashed arrows represent fictitious ("dummy") activities with null execution time, used for specifying pre-requisite relationships, in order to preserve network integrity

- The node event only "occurs" after all the inbound activities finish



Dotted line is 'dummy activity' to ensure that 'Execute change' starts only after both 'Verify change order' and 'Assure stock' are completed.

Representing timing in AOA diagrams

A – Activity

T – Time to complete activity (= EFT-EST = LFT–LST)

EST – Earliest Start Time

EFT – Earliest Finish Time

LST – Latest Start Time

LFT – Latest Finish Time

Algorithm for building AOA diagrams

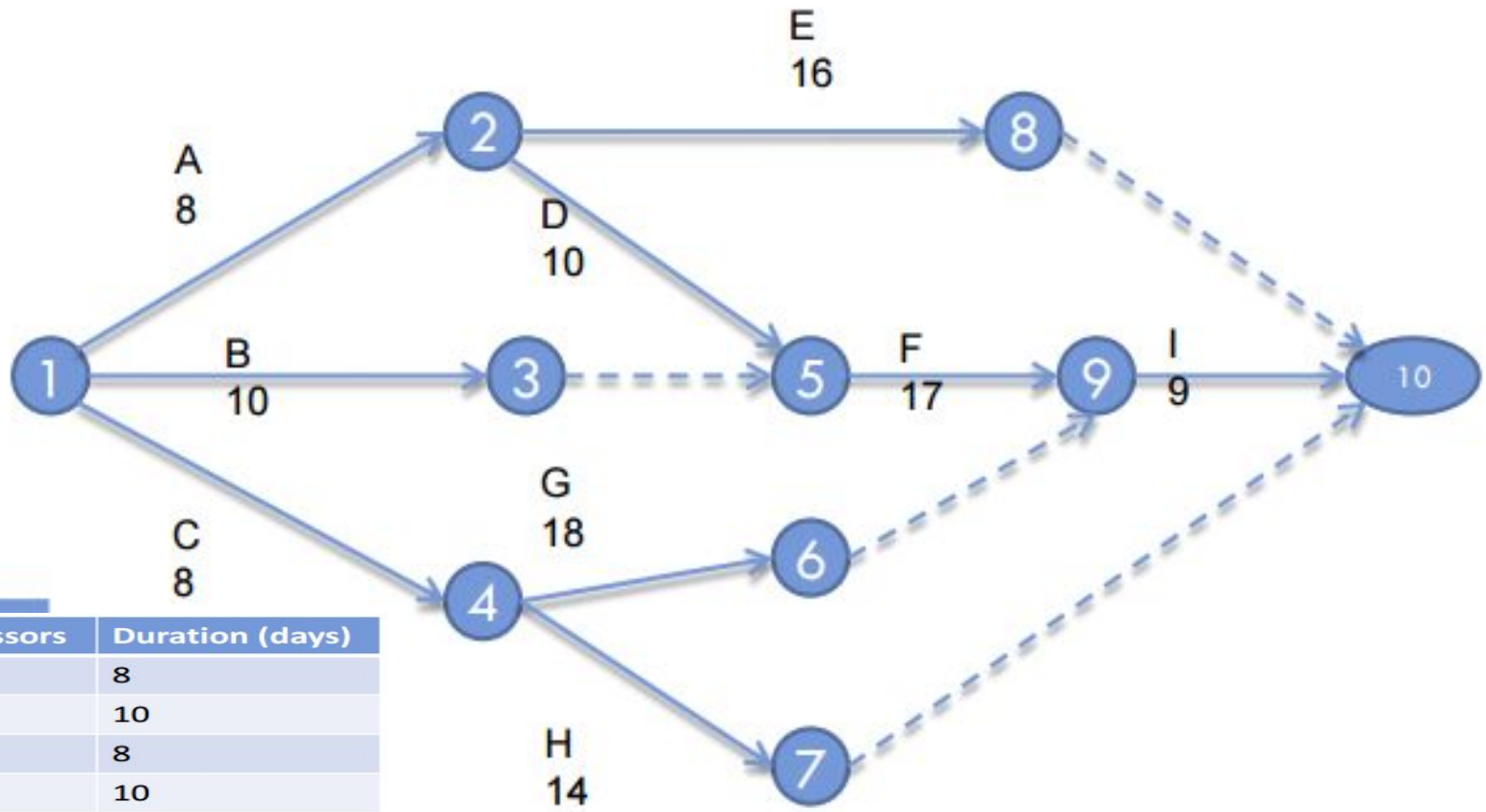
1. Identify and list all activities
2. Assign each activity a unique id
3. Identify and list the dependencies among activities
4. Design a preliminary network
5. Estimate activities durations
6. Add activities durations to the network
7. Compute early start times
8. Compute late start times
9. Fine tune the network
10. Assign resources

Critical path

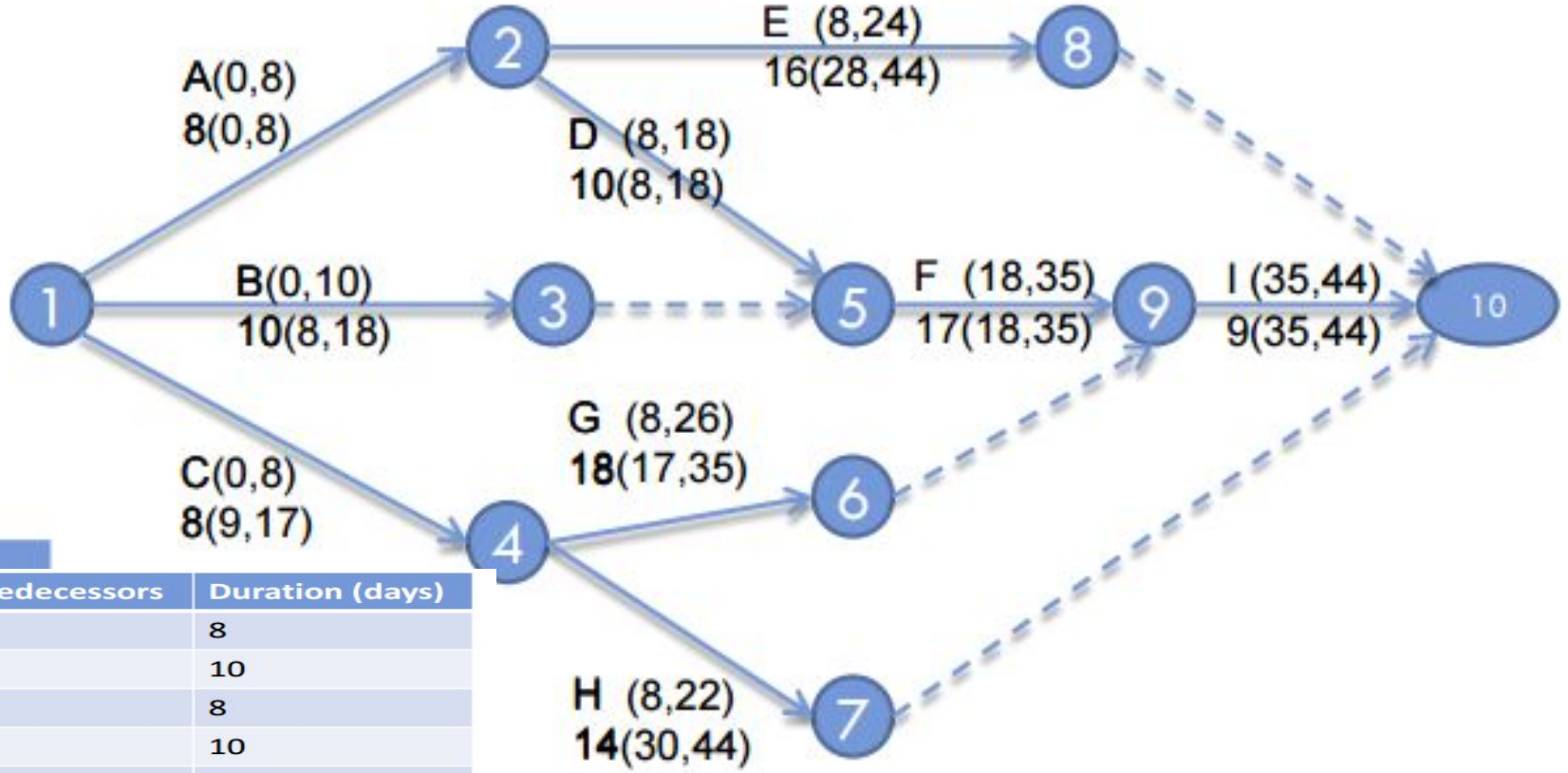
- The longest network path formed by activities where EST = LST
- This is called the **critical path**
  - Any deviation on the duration of activities in this path will have a direct impact on the whole network (i.e. on the whole project schedule)

Create an AOA diagram

Activity	Predecessors	Duration (days)
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9

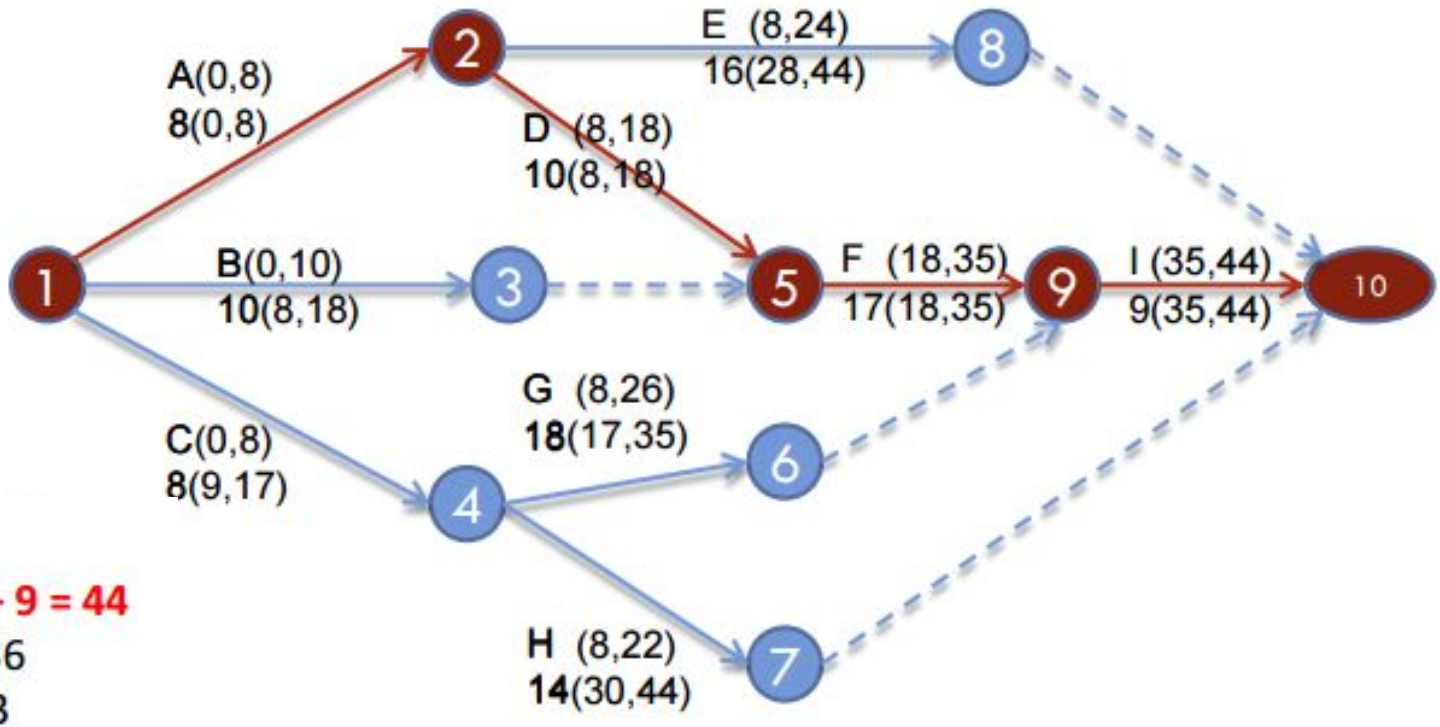


Activity	Predecessors	Duration (days)
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9



Activity	Predecessors	Duration (days)
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9

Critical Path



A-E -> 8 + 16 = 24

A-D-F-I -> 8 + 10 +17 + 9 = 44

B-F-I -> 10 + 17 + 9 = 36

C-G-I -> 8 + 18 + 9 = 33

C-H-> 8 + 14 = 22

Limitations of AOA diagrams

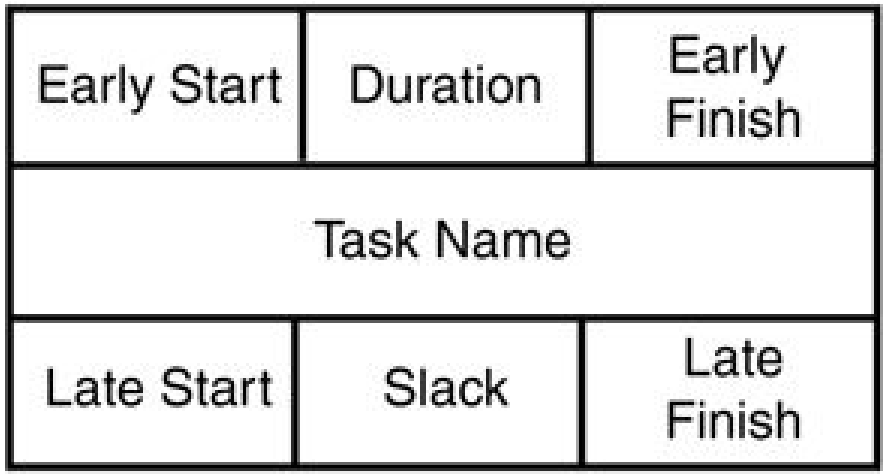
- The need to create “dummy” activities for preserving network integrity
- Only simple dependences can be represented
- Accidental complexity makes this technique hard to apply when projects include many activities

Activity On Node Diagrams

(the Precendence Diagram Method)

Precedence Diagram Method

- Each activity represented by an **activity node**



- **Predecessor/successor (aka dependence)** relationships represented by arrows

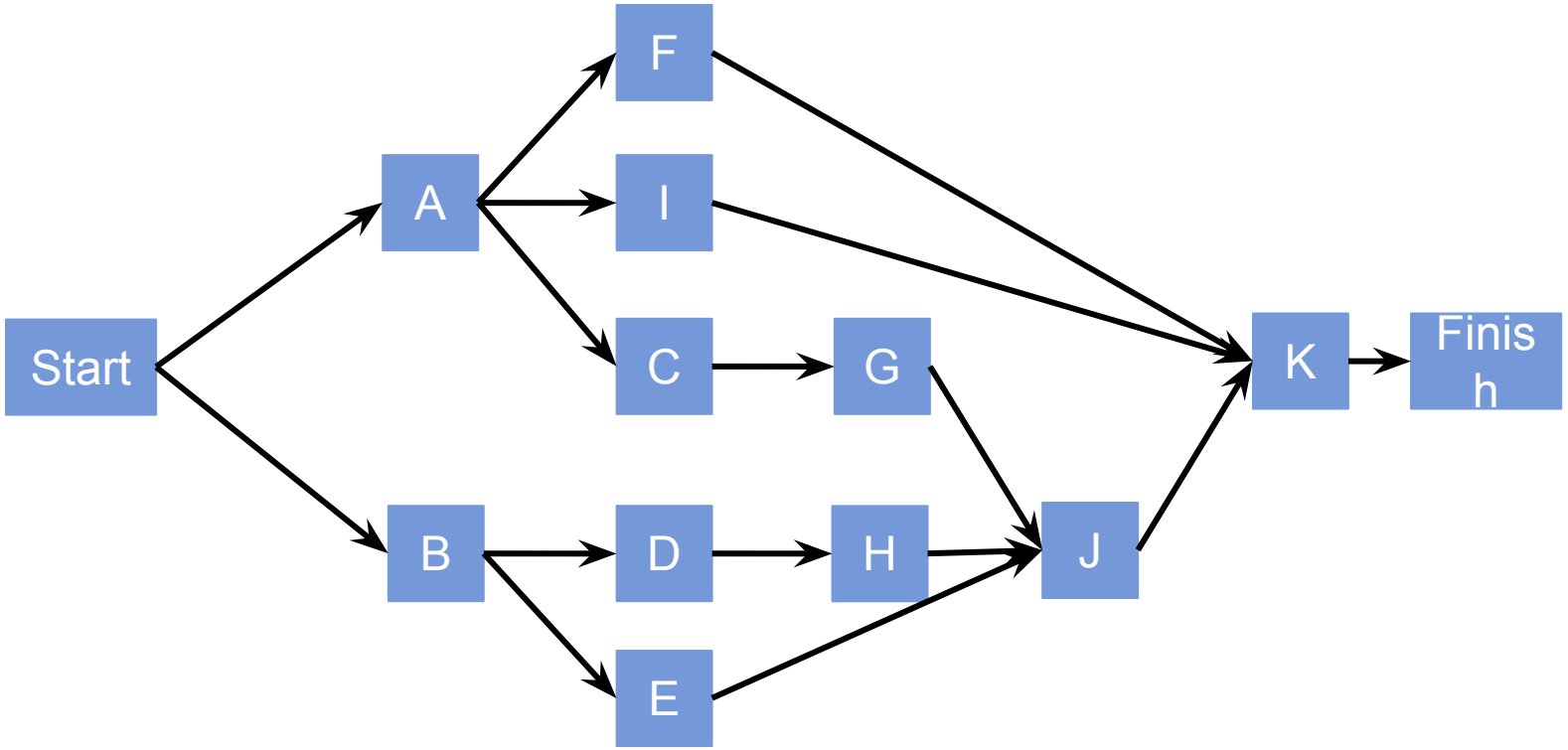
Duration = Early Finish – Early Start = Late Start – Late Finish
Slack = Late Start – Early Start = Late Finish – Early Finish

The computer geeks school creation activities

Activity	Description	Predecessors
A	Select administratives and teachers	-
B	Select location	-
C	Select equipment	A
D	Prepare construction plans and layout	B
E	Bring utilities to site	B
F	Interview applicants and fill positions support and tech staff	A
G	Purchase and receive equipment	C
H	Build the computer geeks school	D
I	Develop information system	A
J	Install the equipment	E, G, H
K	Train staff	F, I, J

Activity on Node diagram

Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J



Example:

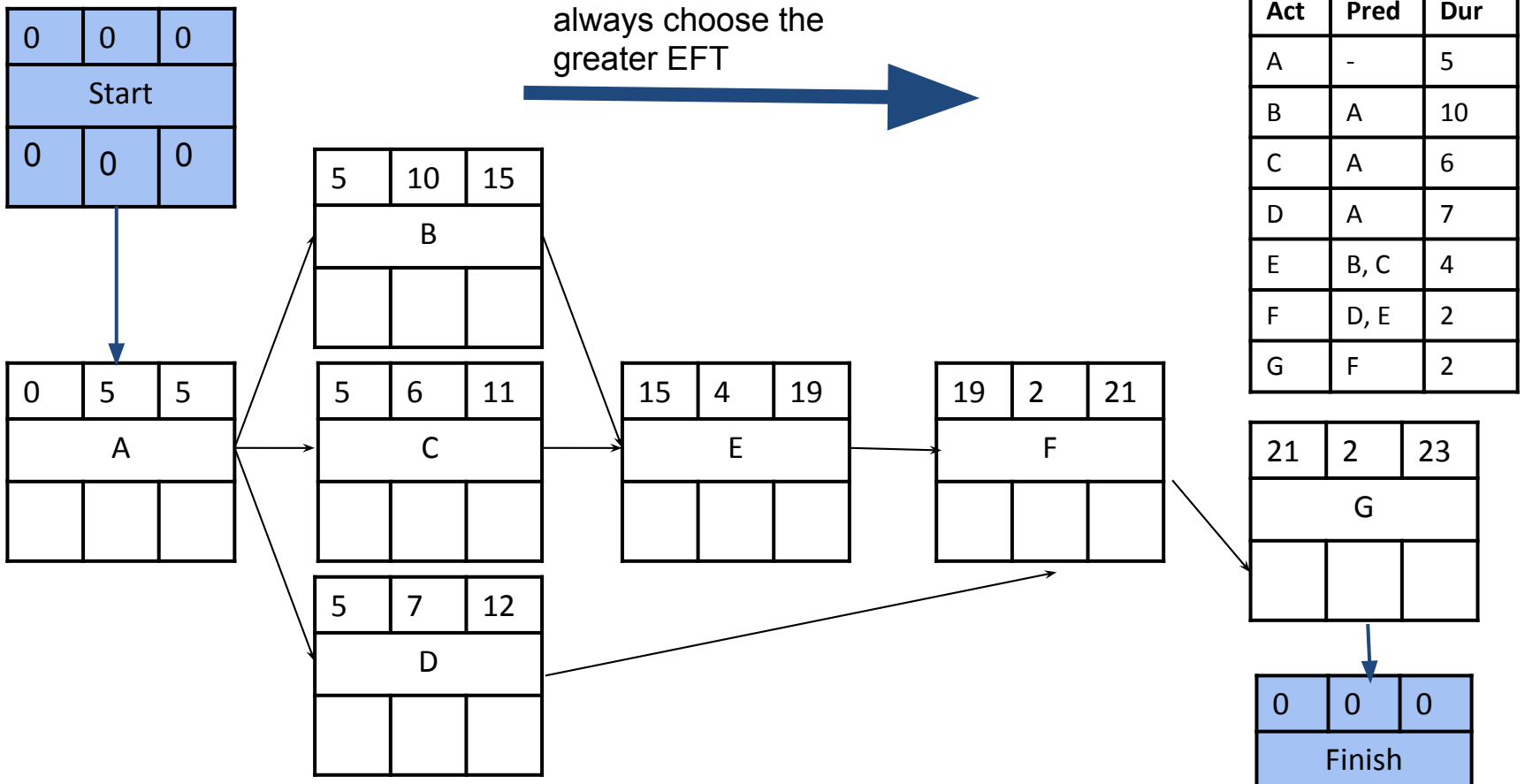
- A company decides to reengineer its IT system.
- They will need new hardware, network and internet access, along with the corresponding software
- The Project Manager has already made a description of the activities to conduct and the time required for each of them

- How much time do we need for this project?

Project data

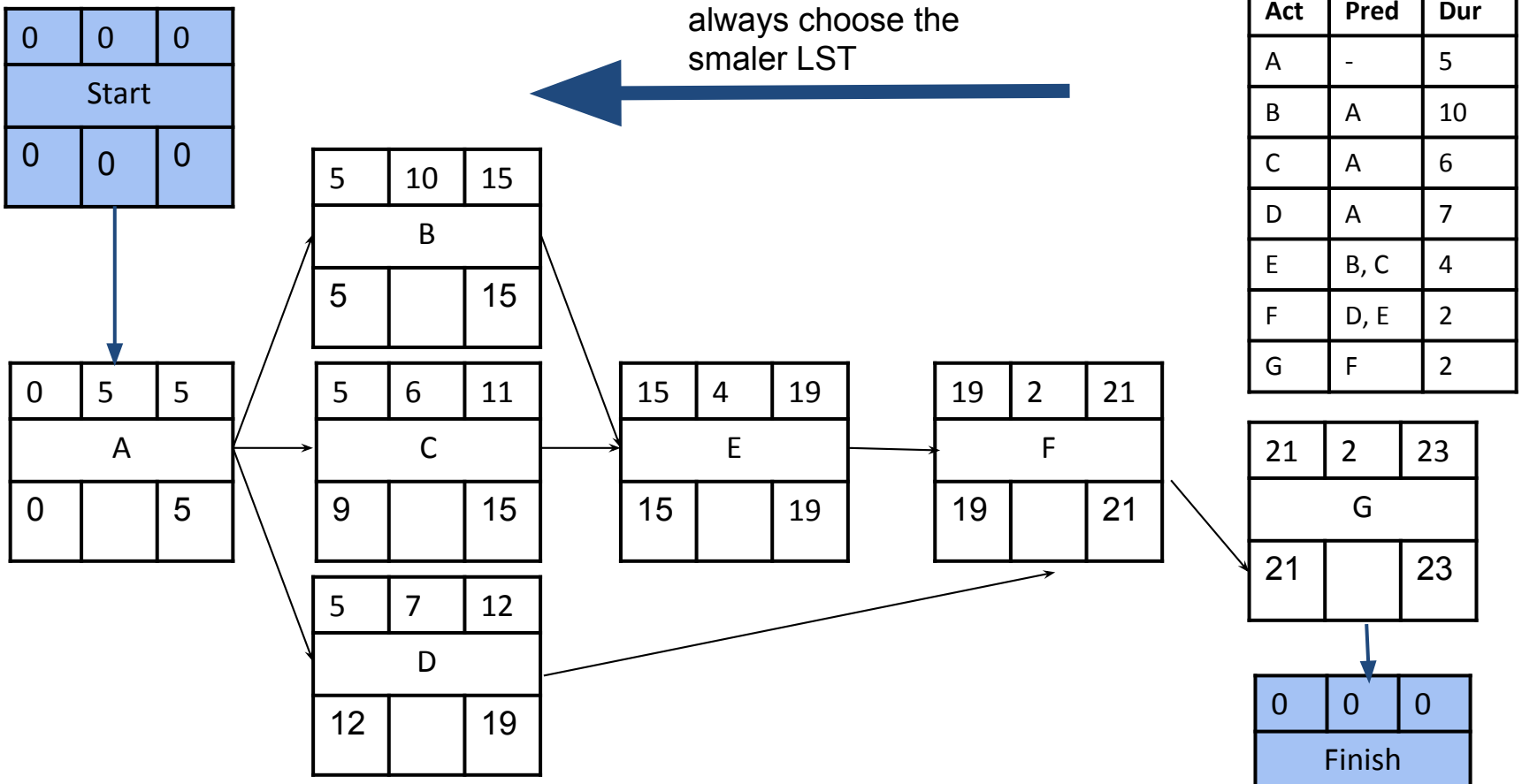
Activity	Predecessors	Duration (in months)
A. Plan project	-	5
B. Acquire hardware	A	10
C. Select location	A	6
D. Determine environmental needs -Power -Phones -Workstations (pcs/laptops) -Software -CASE tools -Furniture	A	7
E. Install hardware	B, C	4
F. Install environmental needs	D, E	2
G. Start production	F	2

How much time do we need for this project?



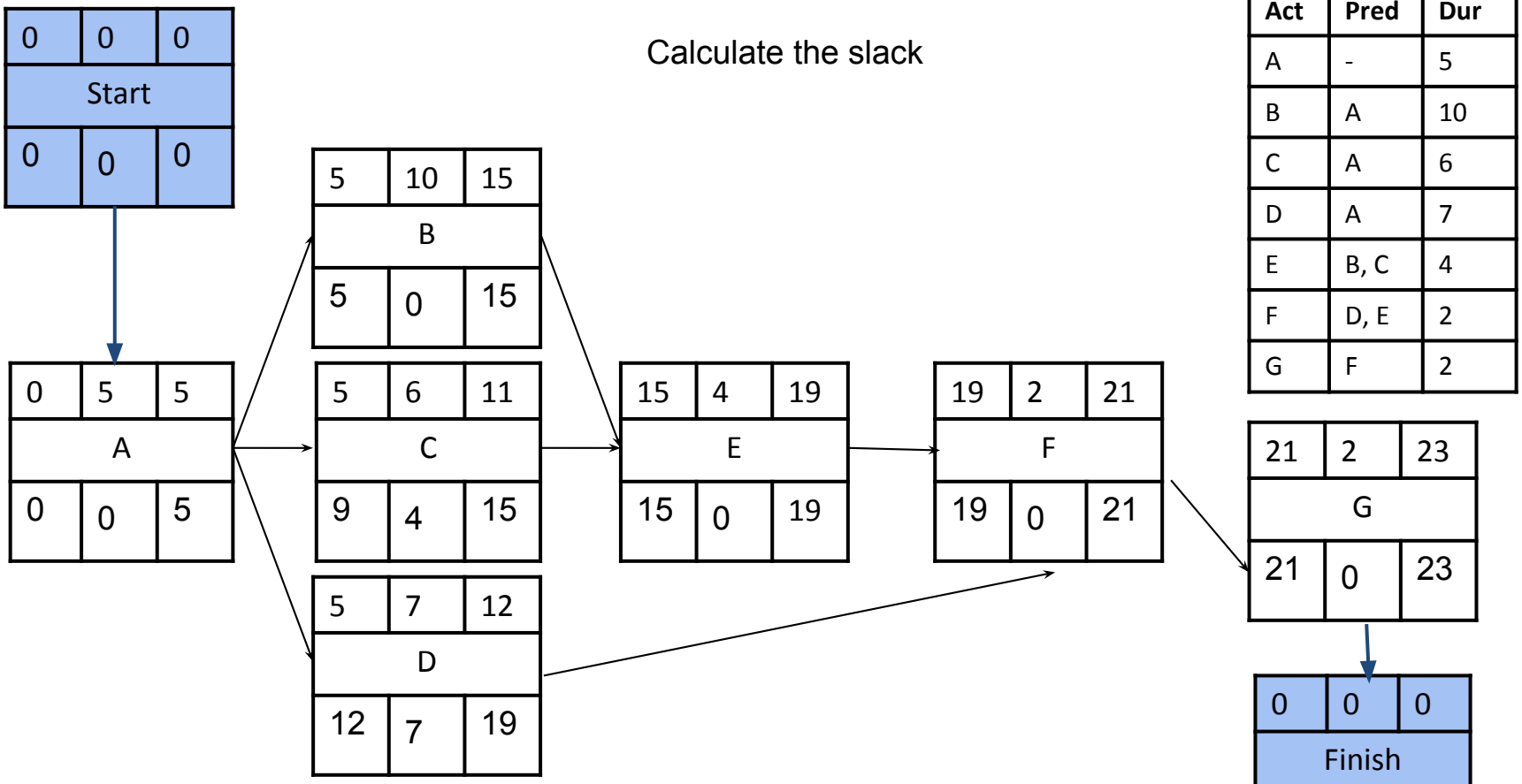
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2

How much time do we need for this project?



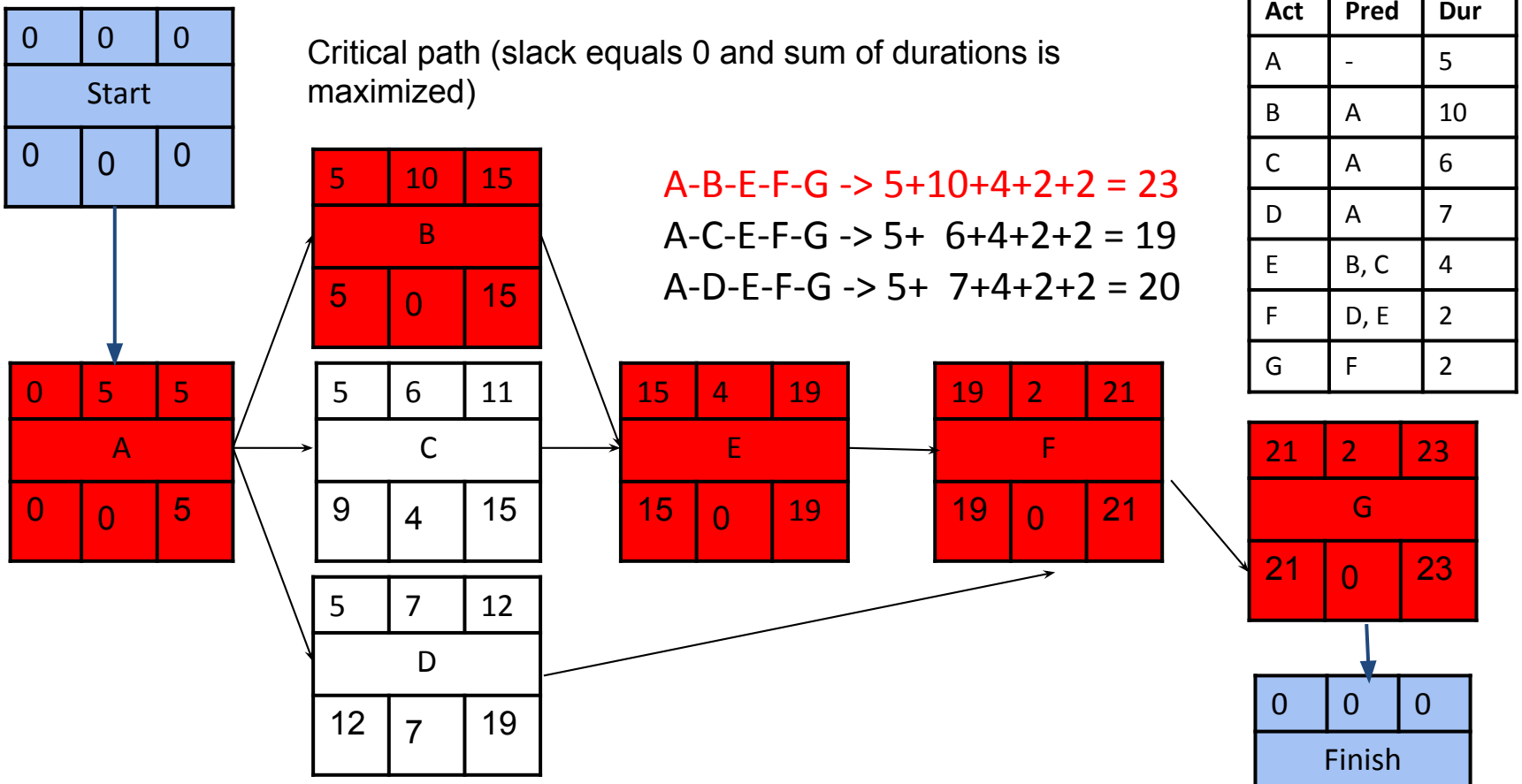
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2

How much time do we need for this project?



Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2

How much time do we need for this project?



Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2

Bibliography

“Gestão de Projectos de Software”, António Miguel, FCA, 3ª Edição 2008