



فرآیند یکپارچه برنامه ریزی، زمانبندی و کنترل پروژه

بخش سوم:

استاندارد اجرایی: شناسایی و تعریف همه فعالیتها

ارائه دهنده:

سید محمدرضا علوی پور

(PhD, PMP, CMIT)

What we will learn?

- ☐ Practical guide for project schedule
- ☐ Capturing and defining all activities
- ☐ Work breakdown structure (WBS)
- ☐ Recommended approach in developing WBS
- ☐ Checklist for a scheduler for capturing all activities
- ☐ An auditor's key questions and documents for capturing all activities

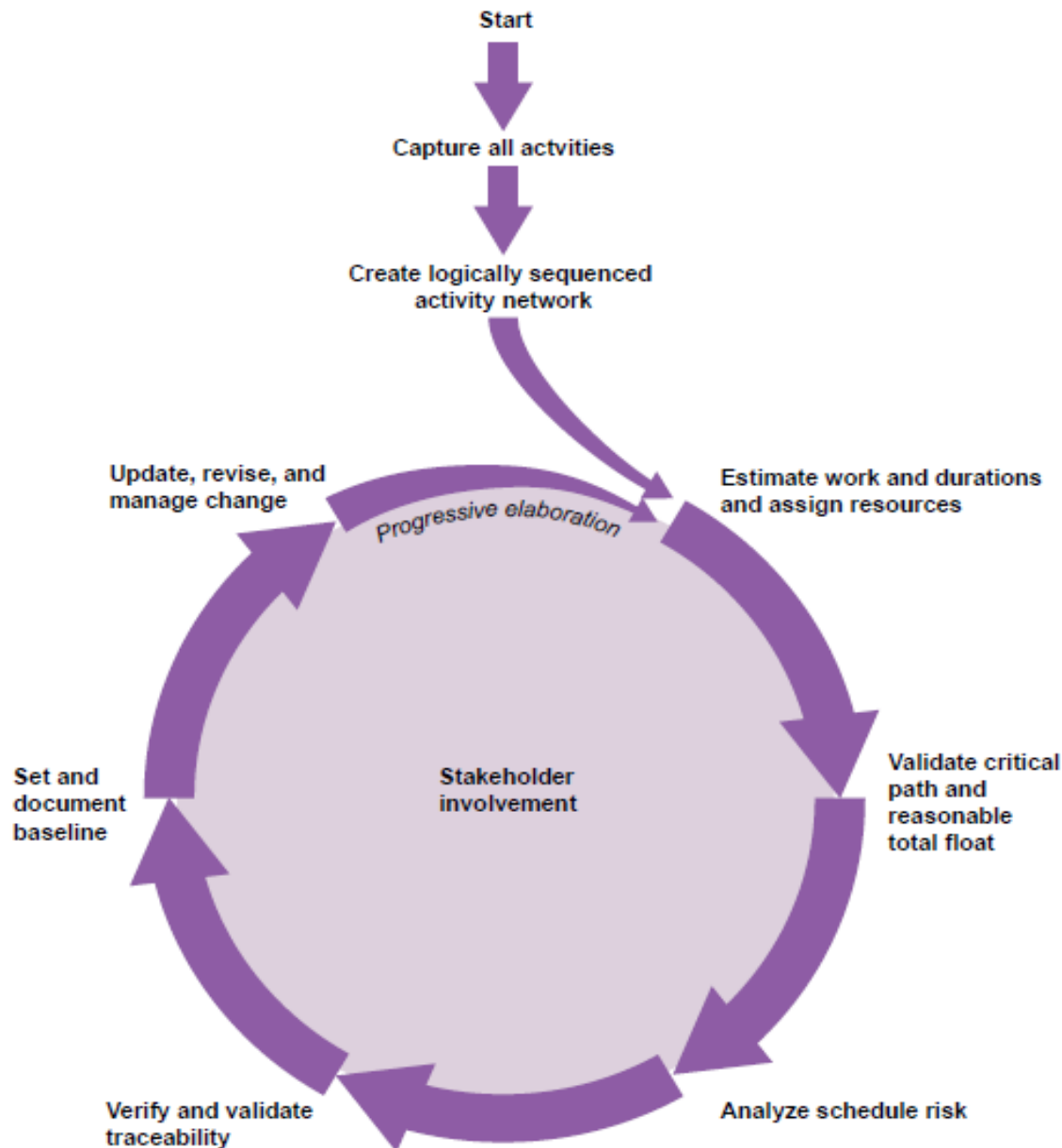
Practical guide for project schedules



فرآیند یکپارچه برنامه ریزی، زمانبندی، ارزیابی و کنترل پروژه استاندارد اجرایی: شناسایی و تعریف همه فعالیتها

A process for creating and maintaining reliable schedules

- ❑ This is a cyclic process
- ❑ Elaborated through rolling wave process.



A process for creating and maintaining reliable schedules

Process step	Corresponding action	Section
Capture all activities	Capturing all activities	3
Create logically sequenced activity network	Sequencing all activities	4
Estimate work and durations and assign resources	Allocating and leveling resources	5
	Determining and assigning durations to activities	6
Validate critical path and reasonable total float	Critical path determination, assessment, and validation	7
	Ensuring reasonable total float	8
Analyze schedule risk	Schedule constructability and risk analysis	9
Verify and validate traceability	Verifying that the schedule can be traced horizontally and vertically	10
Set and document baseline	Setting, monitoring, controlling, and Maintaining a baseline schedule	11
Update, revise, and manage change	Setting, monitoring, controlling, and Maintaining a baseline schedule	11
	Updating the schedule using actual progress and logic	12

Capturing and Defining All Activities

Integrated Master Schedule (IMS) – Apportioned effort activities

PMI CPM Scheduling for Construction Best Practice

- **Apportioned Effort**

If a task has direct, supportive relationship to another task that has its own earned value, the value of the support task may be determined based on (or apportioned to) the earned value of the reference base activity. Examples of proportional tasks include quality assurance and inspection activities.

- **Level of Effort**

Some project activities do not produce tangible outcomes that can be measured objectively. Examples include project management and operating a project technical library. These activities consume project resources and should be included in earned value management planning and measurement. In these cases, the level of effort (LoE) technique is used for determining earned value. A planned value is assigned to each LoE task for each measurement period. This planned value is automatically credited as the earned value at the end of the measurement period.

Integrated Master Schedule (IMS) – Apportioned effort activities

❑ These activities are

1. Performed proportionally to the **main tasks**

2. Documented in a **consistent manner**

and at a **constant and measurable ratio/factor**, to one or more discrete work packages.

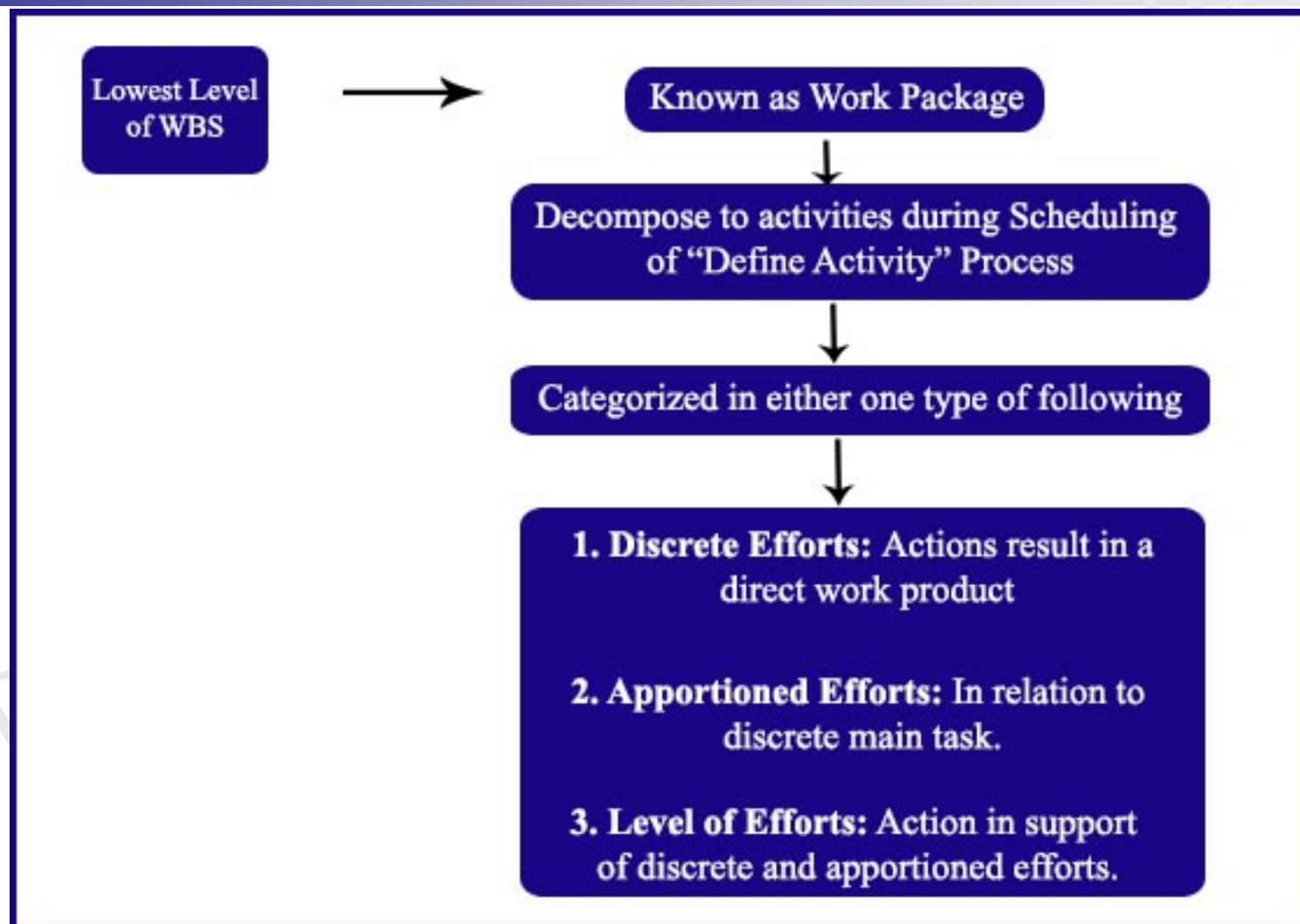
For instance, in a **pipeline construction project**, *Inspection of Weld's activity* is an “Apportioned Effort” activity and a **mainline welding activity** is the “*Discrete Effort*” of it.

LOE is used for **work packages** where

1. Progress **cannot** easily be **measured**

2. Nature of the work is **unsuitable** for using ‘apportioned effort’ to calculate progress.

Integrated Master Schedule (IMS) – Apportioned effort activities



Integrated Master Schedule (IMS) – Constructing an IMS

❑ IMS includes:

1. Level 1 - Summary schedule
2. Level 2 - Intermediate schedule
3. Level 3 - All detailed schedule

❑ Program teams define the information at **each level**.

❑ The **dates** between **each level** of the **schedule hierarchy** should be **vertically traceable**.

❑ All levels of schedule **hierarchy** come from **one data source**, the **detailed IMS**.

❑ The **Summary Master** and **Intermediate level** schedules are simply **summarized roll-ups** of that detailed IMS.

Integrated Master Schedule (IMS) – Summary schedule

❑ May also be called a **Master Phasing Schedule (MPS)**, **Master Plan**, or **Summary Schedule**.

1. The **Program Team** initially **develops** the **program summary master schedule** from the analysis of requirements data.
2. The **program team** **review** and **approve** the **program's top-level schedule**, which serves as a starting point in the **Top Down planning approach**.

Integrated Master Schedule (IMS) – Summary schedule

☐ Summary schedule:

❖ At the **highest level**, the summary schedule provides a **strategic view** of

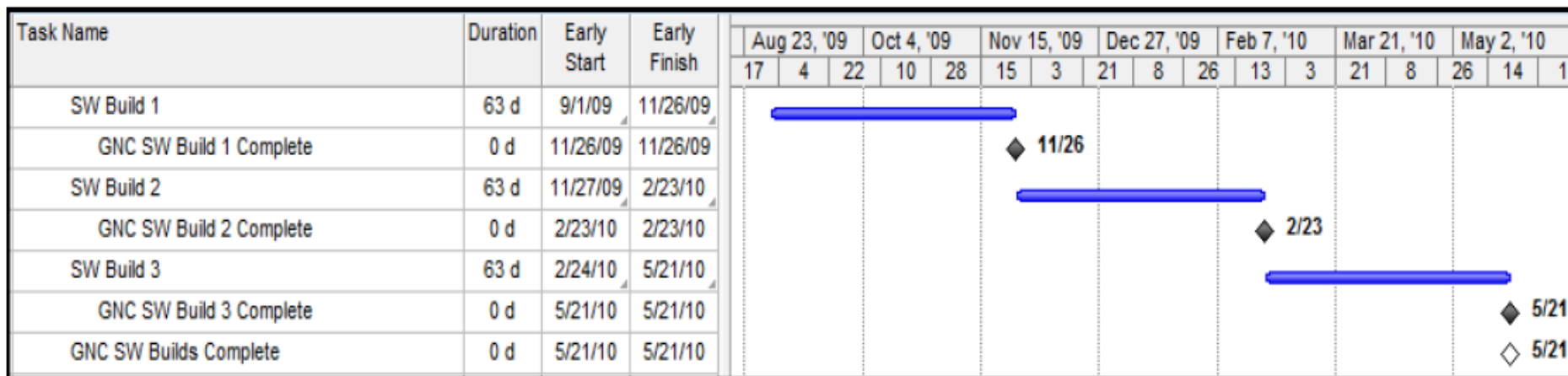
1. Activities
2. Milestones

necessary to **start** and **complete** a program.

☐ **Decision makers and senior management** use summary schedules to **view overall progress** toward **key milestones**.

☐ The dates of **these milestones** are **automatically calculated**.

Integrated Master Schedule (IMS) – Summary schedule



Integrated Master Schedule (IMS) – Intermediate schedule

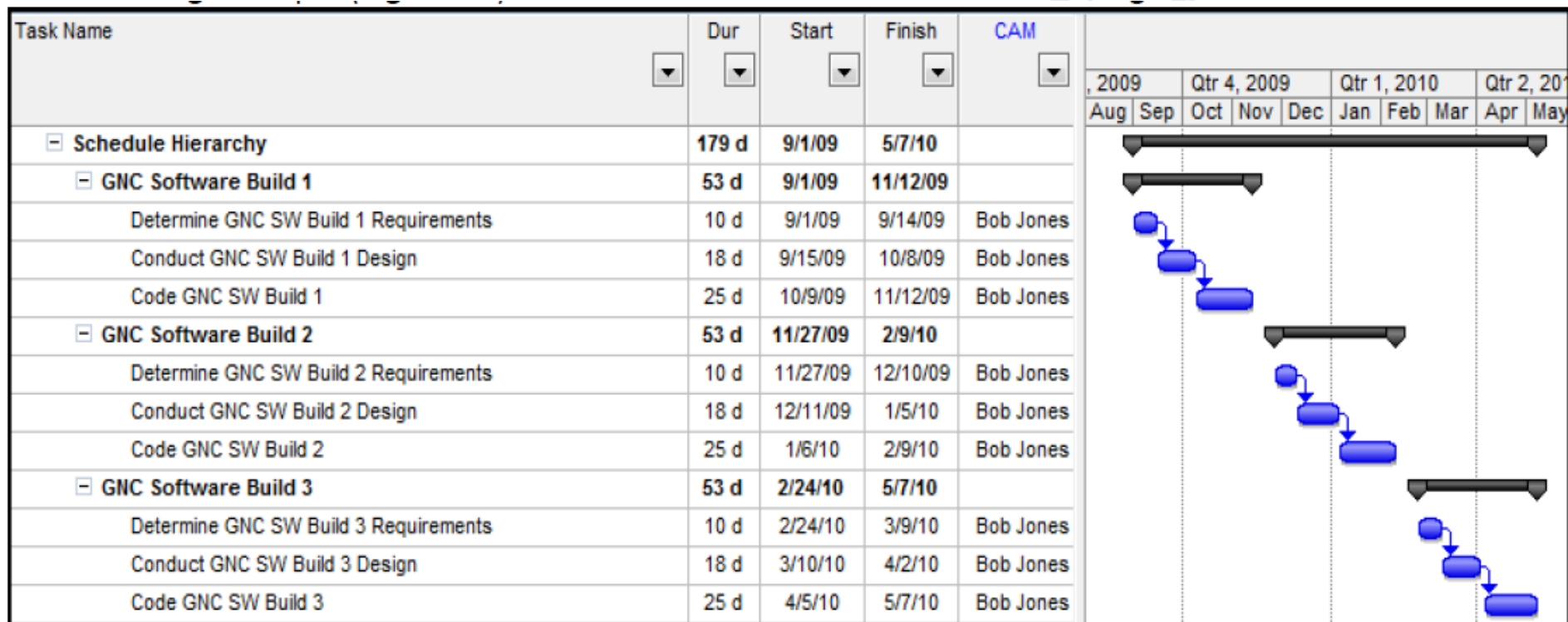
❑ Intermediate schedule includes:

Program activates, milestones, and phases at a level of detail between, but vertically traceable to, the Summary Schedule and Detailed Schedule.

that show the **important steps** in achieving **high-level milestones**.

*For instance, an **intermediate schedule** may show the **interim milestone accomplishments** necessary **before** a major milestone decision or summarized activities related to a **specific trade or resource group**.*

Integrated Master Schedule (IMS) – Intermediate schedule



Integrated Master Schedule (IMS) – Detailed schedule

☐ All detailed schedule:

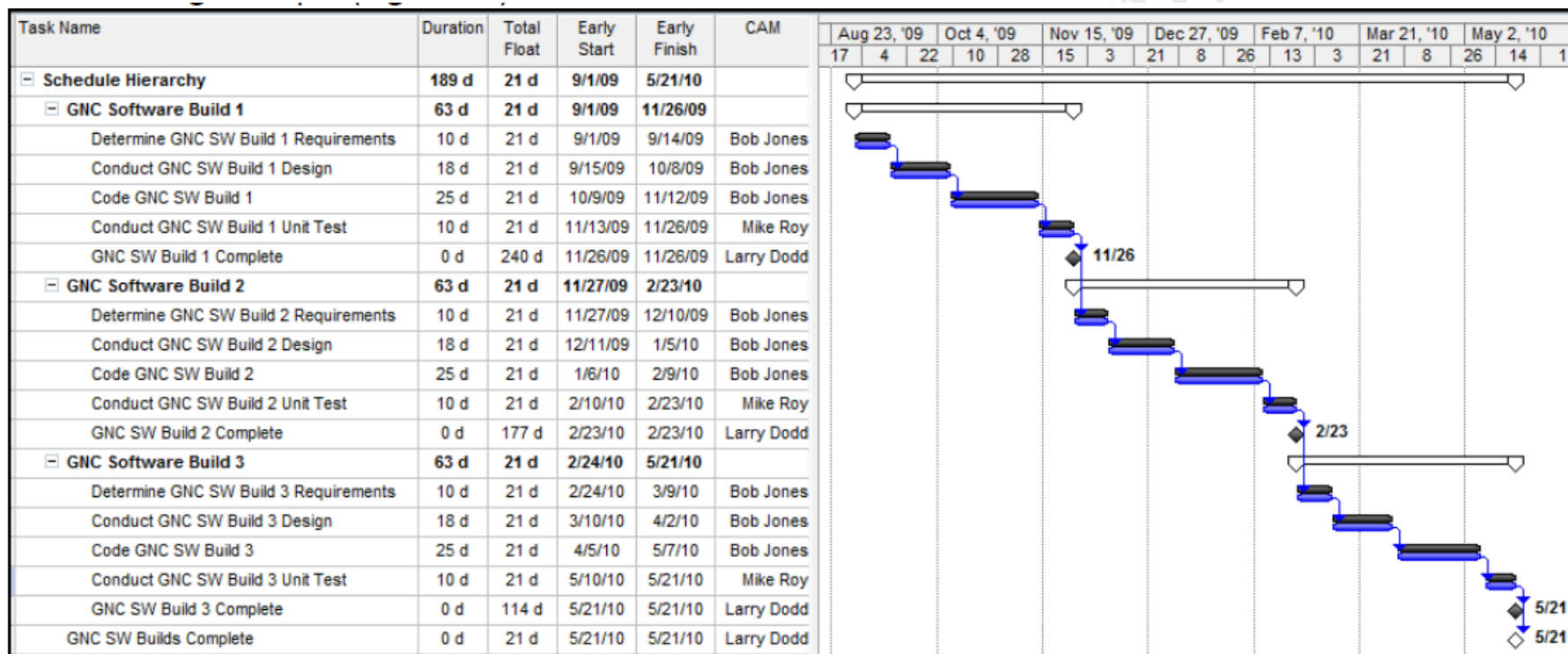
- ❖ At the **lowest level**, the **detailed schedule** lays out the **logically sequenced day-to-day effort** to reach **program milestones**.

- ☐ Delays in lower-level schedules should be **immediately rolled up** to intermediate and summary schedules.

- ☐ Decision makers and senior managers **do not need** to see **detailed schedule**.

- ☐ Specialist contractors need to see the **detailed schedule**.

Integrated Master Schedule (IMS) – Detailed schedule



Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)

□ IMP relationship to the IMS

❖ There are **two common methods** used for creating the IMP to IMS correlation:

1. Use the **IMP Matrix** as the primary architecture / structure of the IMS.

- ❖ This means that the IMP Matrix is the **top three levels** (Event, Accomplishment, Criteria) of the IMS and
- ❖ Detailed tasking below the criteria represents a **further breakdown** of the scheduled work.
- ❖ Additionally, the unique IMP number system is **flowed down** to the **discrete tasks** and maintained in the IMS in field defined by the user.

Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)

❑ IMP relationship to the IMS

❖ There are **two common methods** used for creating the **IMP to IMS correlation**:

1. Use the **IMP Matrix** as the **primary architecture / structure of the IMS**.

IMP

E	A	C	Description
A			Event #01
A	01		Accomplishment #01
A	01	01	Criteria #01
A	01	02	Criteria #02
A	02		Accomplishment #02
A	02	01	Criteria #01
A	02	02	Criteria #02

IMS

E	A	C	T	Description	01	02	03	04	05
A				Event #01					
A	01			Accomplishment #01					
A	01	01		Criteria #01					
A	01	01	01	Task #01	△	△			
A	01	01	02	Task #02		△	△		
A	01	02		Criteria #02					
A	01	02	01	Task #01			△	△	
A	01	02	02	Task #02				△	△

Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)

□ IMP relationship to the IMS

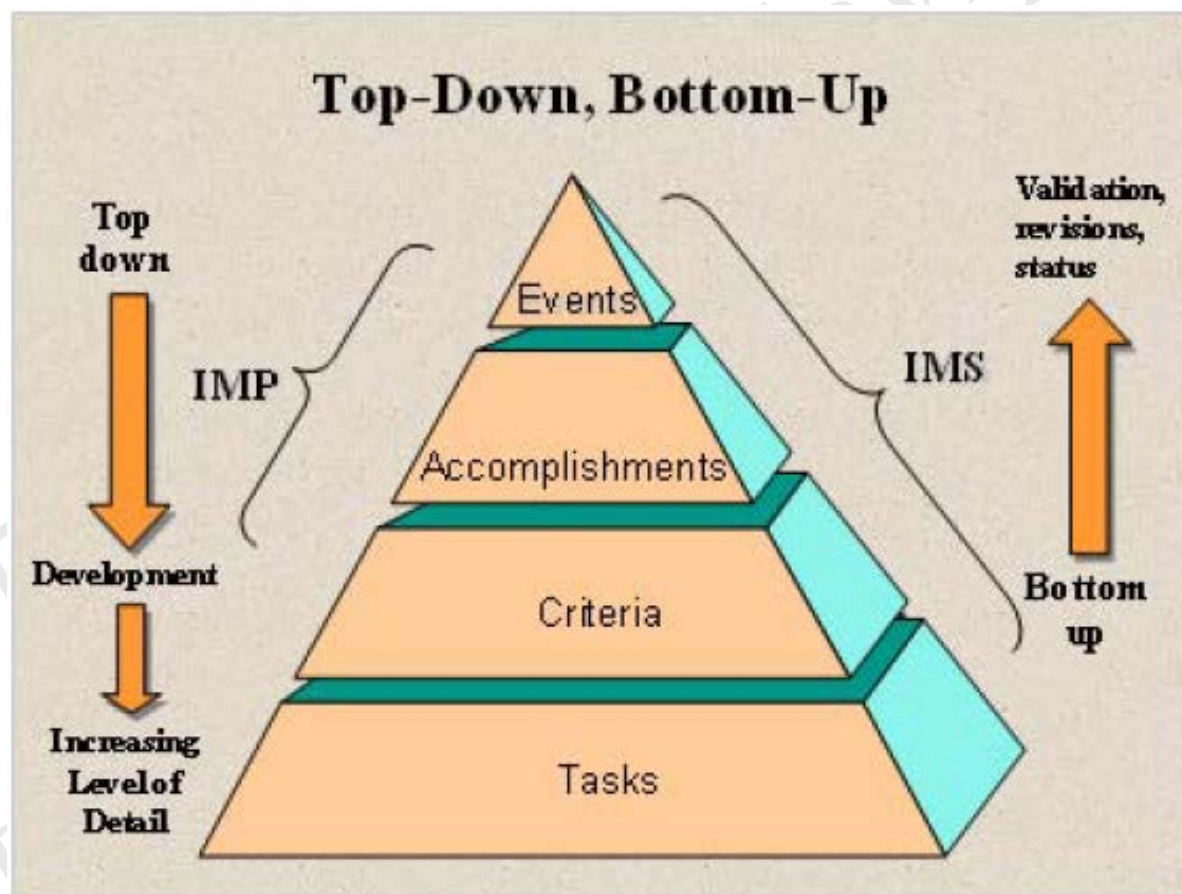
❖ There are **two common methods** used for creating the **IMP to IMS correlation**:

2. **Use the WBS or other structure as the primary architecture of the IMS**

❖ Then **map** the **detailed tasks** and **milestones** (as applicable) to the **IMP Matrix** via a **unique IMP numbering system** in a field defined by the user.

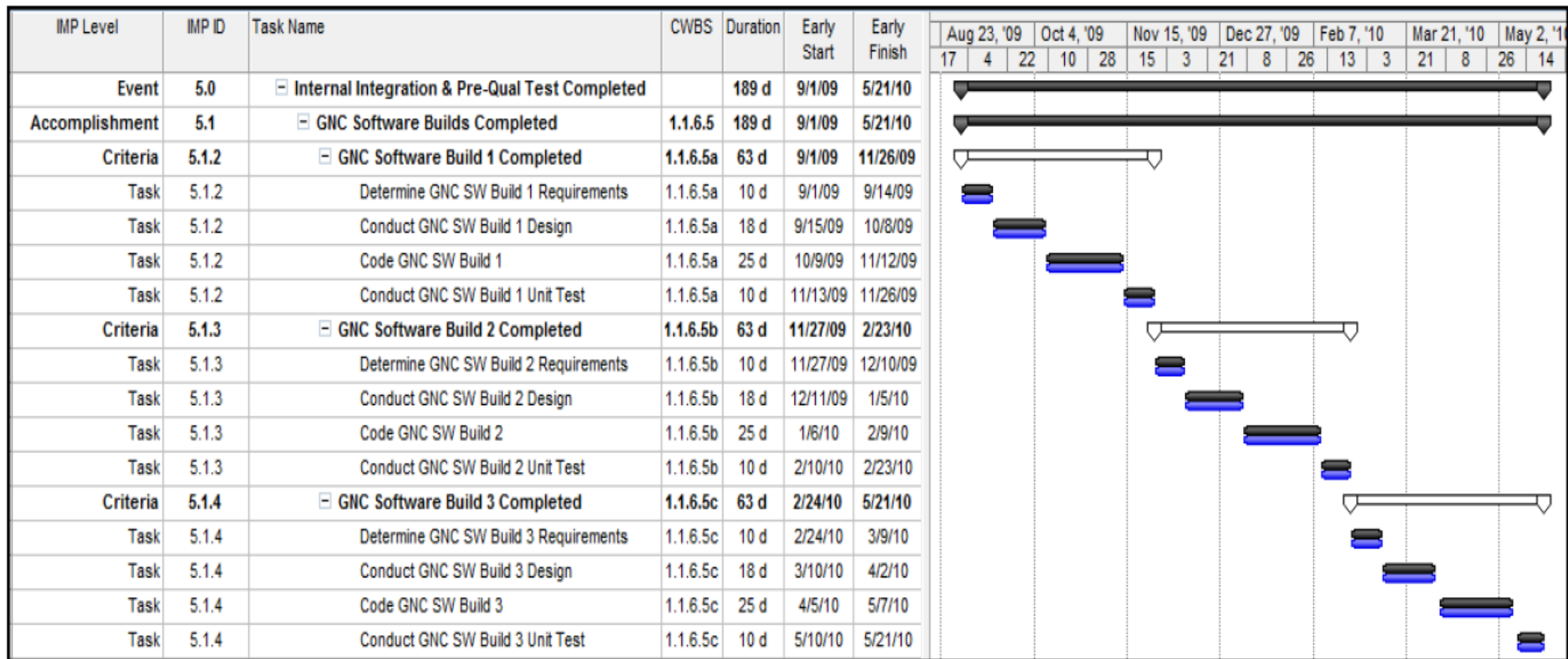
Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)

❑ *Top-down vs. Bottom-up methodology*



Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)

IMS using an IMP



Note: The WBS and IMP structure will not always follow each other 1-to-1 as depicted in this example.

Work Breakdown Structure (WBS)

Introduction

- ❑ WBS provides the **framework** for **organizing** the **tasks** to complete an **entire project**.
- ❑ WBS is a **systematic way** of describing the **components** of a **project schedule**.
 - ❖ *For example, in a typical building the WBS would contain such systems as the site work; structure; mechanical, electrical, and plumbing (MEP); interior finishes; and roofing system.*

PMI CPM Scheduling for Construction Best Practice

3.4.1 Work Breakdown Structure

The work breakdown structure (WBS) is a tree structure, but can also be in an outline, which shows a decomposition of effort required to achieve an objective such as a program, project, or contract. In a project or contract, the WBS is developed by beginning with the end objective and successively decomposing it into manageable components in terms of size, duration, and responsibility (e.g., systems, subsystems, components, tasks, subtasks, and work packages), including all steps necessary to achieve 100% of the scope (the objective).

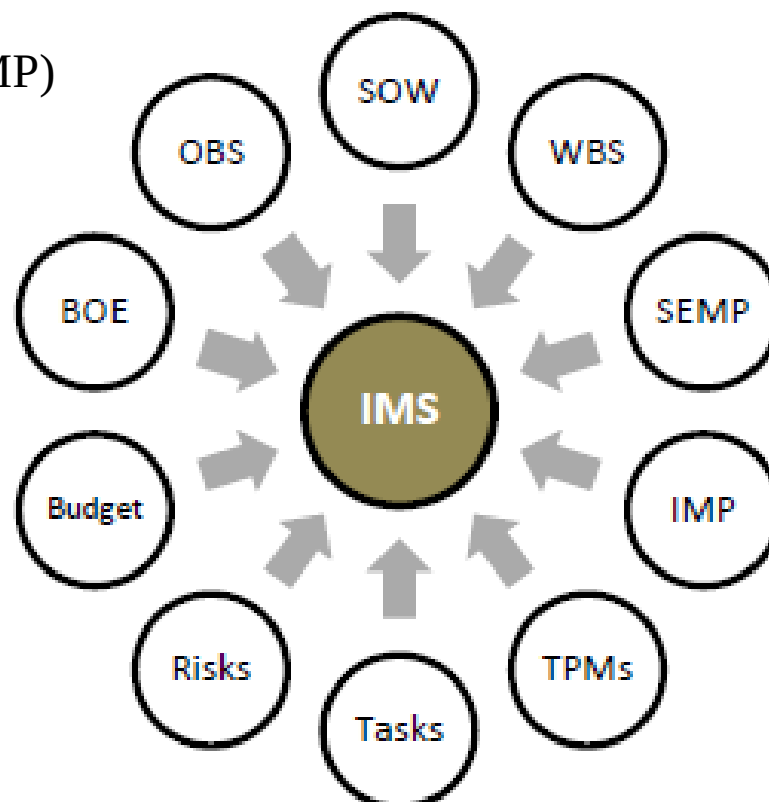
Why we need WBS?

♦ Definition

"A hierarchical decomposition of the total scope of work to be carried out by the project team to accomplish the project objectives and create the required deliverables." (Project Management Institute, 2017)

Why we need WBS?

- ☐ Basis of estimate (BOE)
- ☐ Statement of work (SOW)
- ☐ System engineering management plan (SEMP)
- ☐ Technical performance measure (TPMs)
- ☐ Organizational breakdown structure (OBS)



What else?

Introduction

❑ The **use of a WBS** for organizing project scope has become **more common recently**, particularly as more government projects have required it for the **implementation of earned value management (EVM)**.

❖ Discrete effort, apportioned effort and level of effort (LOE) are the 3 types of activities used in **EVM**.

PMI CPM Scheduling for Construction Best Practice

The WBS step helps accomplish the following key objectives:

- Develops an objective, rational view of the amount of work required
- Helps the team grasp the skills and amount of resources required for the project
- Provides a clear framework for assigning a clear task definition to individuals and for delegating the responsibility for completion
- Lays a foundation for analyzing the task dependencies, and for isolating and managing risks
- Lays a foundation for developing a bottom-up estimate for the project schedule
- Allows tradeoffs to be made consciously and with the proper consensus involved

How to develop?

Introduction

- ❑ The **WBS** is derived from the **statement of work (SOW)** and provides a **common framework** for the natural development of the overall planning and control of a contract.

Use WBS ...

- As the main schedule planning tool
- For identifying project activities
- As a basis for time and cost performance measurement and reporting.

Statement of work (SOW)

- ❑ The **Statement of Work (SOW)** is **formal document** that capture and define the **work activities**, **deliverables** and **timeline a vendor** will execute against in performance of specified work for an owner.
 - ❖ The **SOW** provides insight to **data** that should be included in the **IMP** and the **IMS**.
- ❑ **Detailed requirements** for **pricing** are usually included in the **SOW**, along with **standard regulatory** and **governance terms and conditions**.
- ❑ The **contractor** or **subcontractor** uses the **SOW** to expand the **WBS to lower levels** of detail and to **develop the IMS**.
 - ❖ **Activities** within a **contractor's project schedule** should be **directly traceable** to the **program's SOW**.



Statement of work (SOW)

1 Introduction

Start with what is being done and who is doing it.

2 Purpose

What is the reason for the project?
Explain such in a purpose statement.

3 Scope

What needs to be done in the project,
and what process will be used to do it?

4 Where

Note where the project will take
place: on site, remotely, etc.

5 Tasks

Break down the scope of the project
into smaller tasks that are needed to
complete the work.

6 Milestones

Note the phases of the project as milestones
to break up the larger schedule.

7 Deliverables

List and explain what is due and when
it is due.

8 Schedule

Including due dates of deliverables, add relevant
timelines so that it ends with a deadline.

9 Standards & Testing

Note what if any standards are required and if
there will be a testing stage to the project.

10 Success

Define what the sponsor and/or
stakeholder define as a successful project.

11 Requirements

Determine if there is special equipment
necessary for the project or if the team needs
certifications.

12 Payments

How will pay be delivered and on what
schedule, after milestones, cyclical?

13 Other

Note what didn't fit in previous categories, such
as if there are security issues, travel pay, etc.

14 Close

Lastly, list how deliverables will be delivered, as
well as signing off and archiving all paperwork.

Development of work breakdown structure (WBS)

- ❑ Begins with the **definition** of the **major systems** or **components** of a project.
- ❑ **Then** each system is defined in greater and **greater detail** until there exists a **discrete or measurable piece of work** and a **single responsibility**.
 - ❖ These tasks are **work packages**.
 - ❖ **Each work package** is a **single discrete scope of work** that is to be performed by a **single responsible party**.
 - ❖ **Each work package** has a **definite starting** and **ending point**.
- ❑ The **level of detail** should be **carefully selected** and will **depend** primarily on the **size**, **type**, and **needs** of the project.

Development of work breakdown structure (WBS)

- ❑ **Information** should be provided by **various individuals**, such as the **project manager**, **job superintendent**, and **others**.
- ❑ **Development** of the WBS should be from the **top down and not from the bottom up**.
- ❑ As **information is gathered**, also **collect cost/budget information and schedule data**.
- ❑ The **WBS** will be used to **provide** further definition to the **tasks to be performed**.
 - ❖ *For example, it will serve as the **basis for determining the task durations** and to estimate the costs.*

***WBS** becomes the **backbone** of the project control or project tracking system.*

***WBS** ultimately serves as a **benchmark** for measuring performance, and it also serves as the historical database for the project.*

Development of work breakdown structure (WBS)

- ❑ WBS needs **not** to be in any **particular order**.
- ❑ WBS should **not** be **confused** with the **schedule**.
 - ❖ WBS will be **utilized** to **develop the schedule**.
- ❑ There should be **only one WBS** for **each project**.
- ❑ WBS of the project should **match the WBS** used for the cost estimate, schedule, cost control, and job cost ledger, etc.

Development of work breakdown structure (WBS)

*Remember that the **WBS** documents the scope of work involved in the project but **not the plan of how** the project will actually be constructed.*

*WBS and schedules are **progressively elaborated**.*

*WBS and schedules should **systematically be interrelated**.*

Development of work breakdown structure (WBS)

WBS Should
Decompose the entire project scope
Include revisions for authorized changes
Be the common reference for project reporting
Use parent-child coding system to identify WBS elements
Be changed only through formal change control procedures
Provide a breakdown structure that is meaningful to those performing the work
Be used to identify project activities

WBS Should Not
Be a flat listing of elements
Be a list of activities
Provide a chronological listing of work
Contain redundant elements
Contain heterogeneous WBS elements at the same level
Present information at an improper level of detail or breakdown
Stay static if the project scope changes
Be developed or updated independently from the project schedule

Recommended Approach in developing Work Breakdown Structure (WBS)

Introduction

❑ There is no single correct solution.

- ❖ Individuals with **different experiences** will simply use **different approaches**
 - ✓ This **does not present problems** as long as the project is **properly scoped with the WBS.**

Use the following two types of WBS effectively:

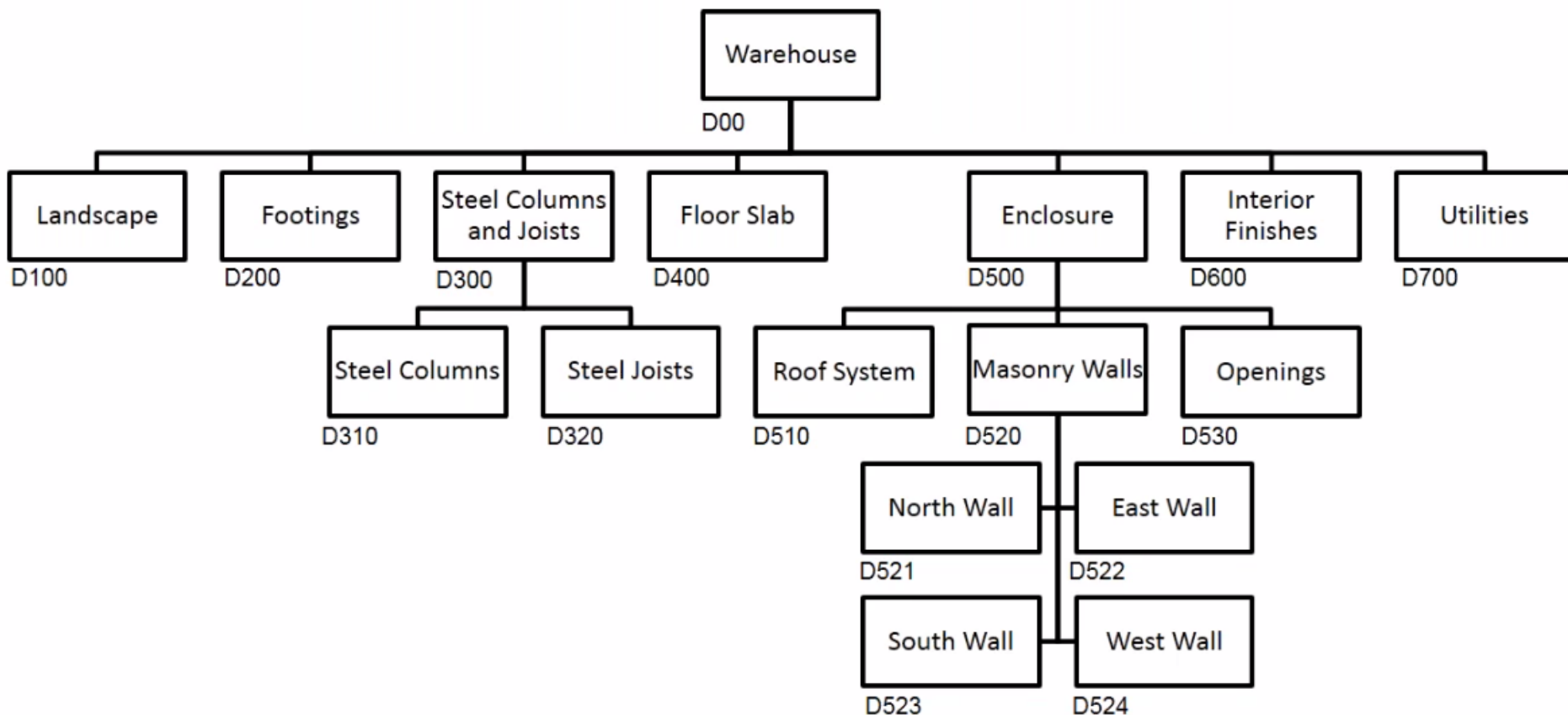
Deliverable-oriented WBS

- A deliverable-oriented WBS defines the project's scope of work in terms of deliverables and decomposes these deliverables into components.

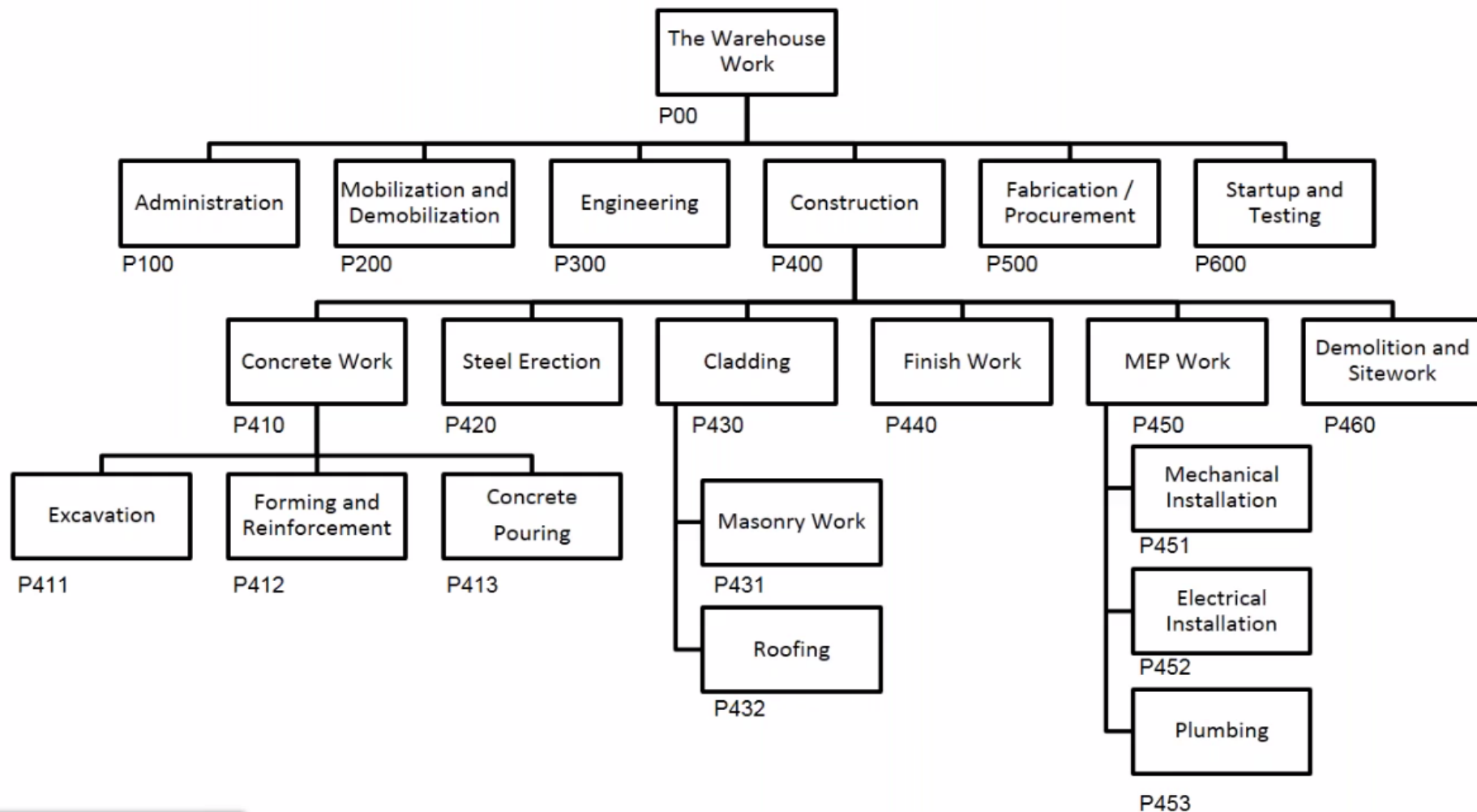
Process-oriented WBS

- A process-oriented WBS defines the project's scope of work in terms of process steps or work phases.

Deliverable or product oriented WBS – Example

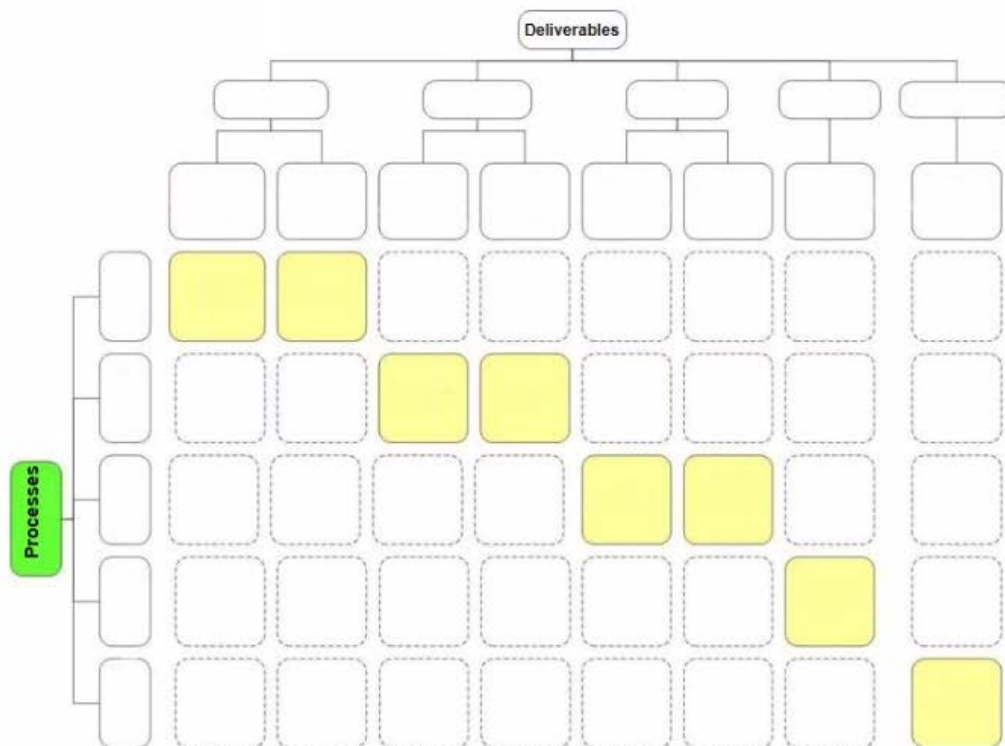


Process oriented WBS – Example



WBS integration matrix

- ◆ Work Done on an Object = Acting Force * Displacement
 - A WBS integration matrix is created by integrating deliverable- and process-oriented WBS's

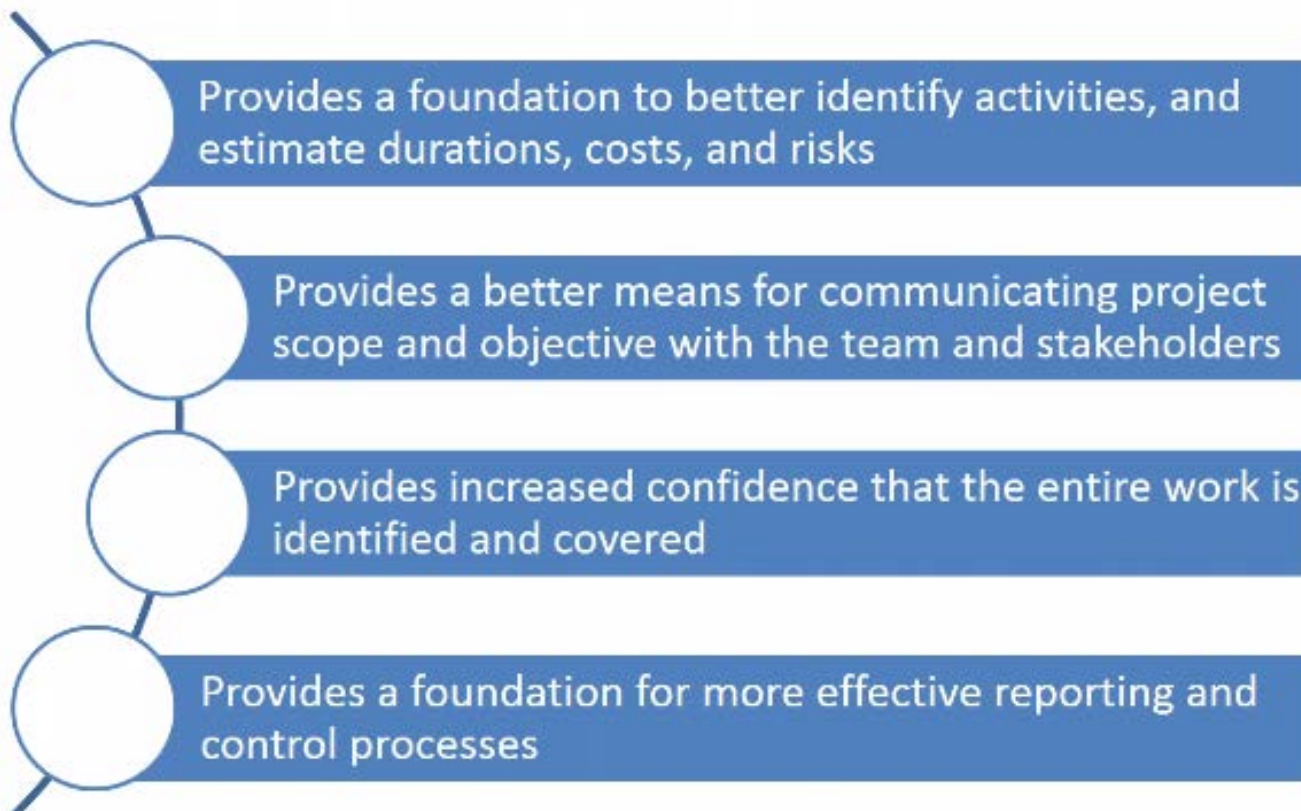


WBS integration matrix

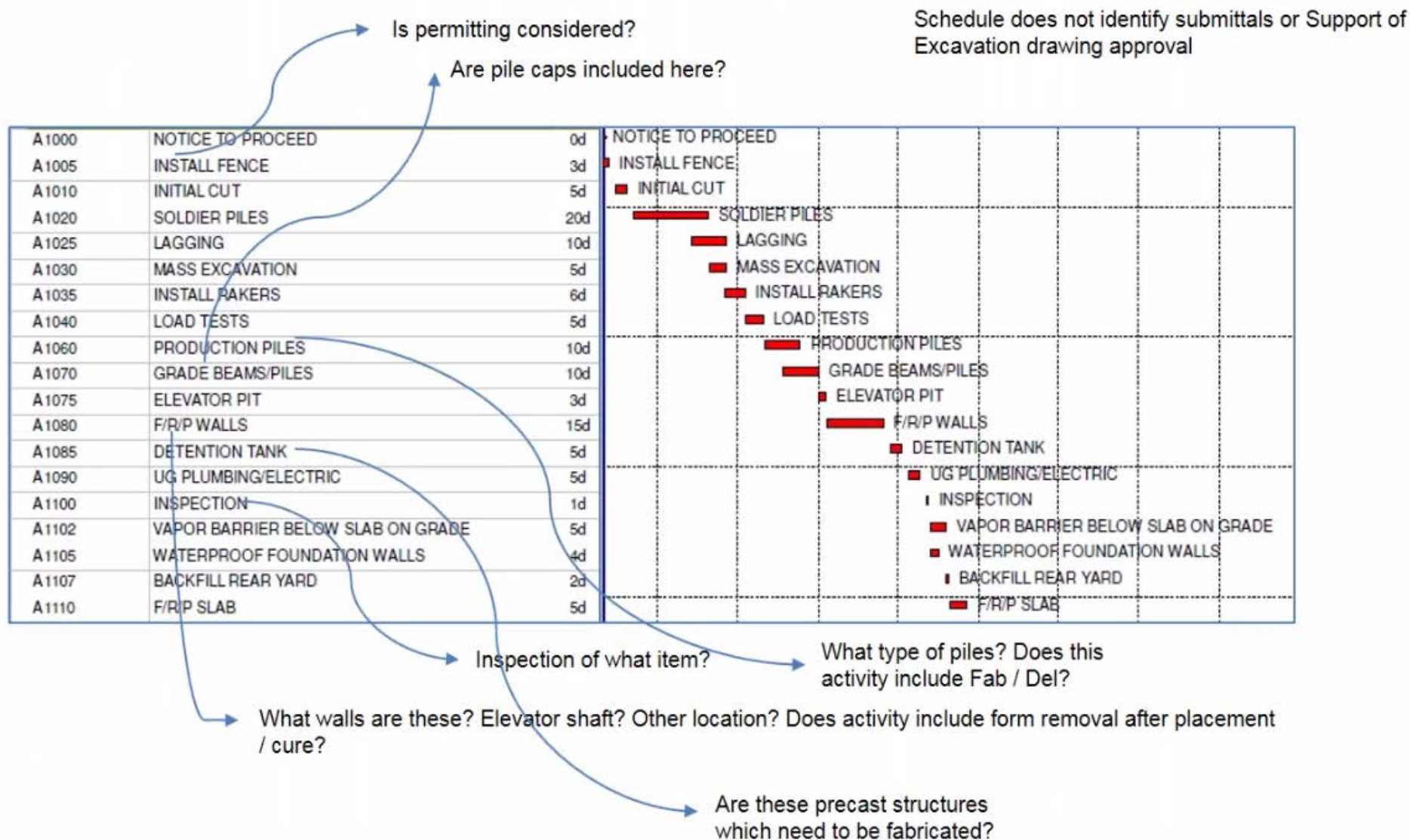
	ID	Process - Oriented WBS Deliverable - Oriented WBS																				DI	
		P00	P100	P200	P300	P400	P410	P411	P412	P413	P420	P430	P431	P432	P440	P450	P451	P452	P453	P460	P500	P600	
			The Warehouse Work	Administration	Mobilization and demobilization	Engineering	Construction	Concrete Work	Excavate	Form and Reinforce	Concrete Pouring	Steel Erection / Install	Cladding	Masonry Work	Roofing	Finish Work	MEP Work	Mechanical Installation	Electrical Installation	Plumbing	Demolition and Sitework	Fabrication / Procurement	Startup and Testing
D00	Warehouse		R01	R02																		R01	
D100	Landscape				R03															R16			
D200	Footings				R03			R07	R07	R07													
D300	Steel Columns and Joists				R03																R17		
D310	Steel Columns											R08											
D320	Joists											R08											
D400	Floor Slab							R07															
D500	Enclosure																						
D510	Roof System				R04										R11						R17		
D520	Masonry Walls				R04											R10							
D521	North Wall														R10								
D522	East Wall														R10								
D523	South Wall														R10								
D524	West Wall														R10								
D530	Openings				R04						R09											R18	
D600	Interior Finishes				R05											R12						R19	
D700	Utilities				R06													R13	R14	R15		R20	

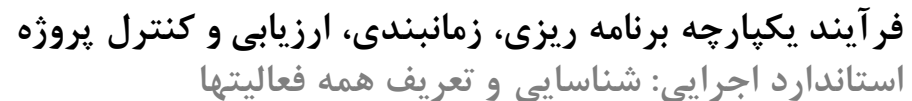
WBS integration matrix

❑ Methodology



WBS integration matrix – Case study: Existing schedule (Example)



[illegible]

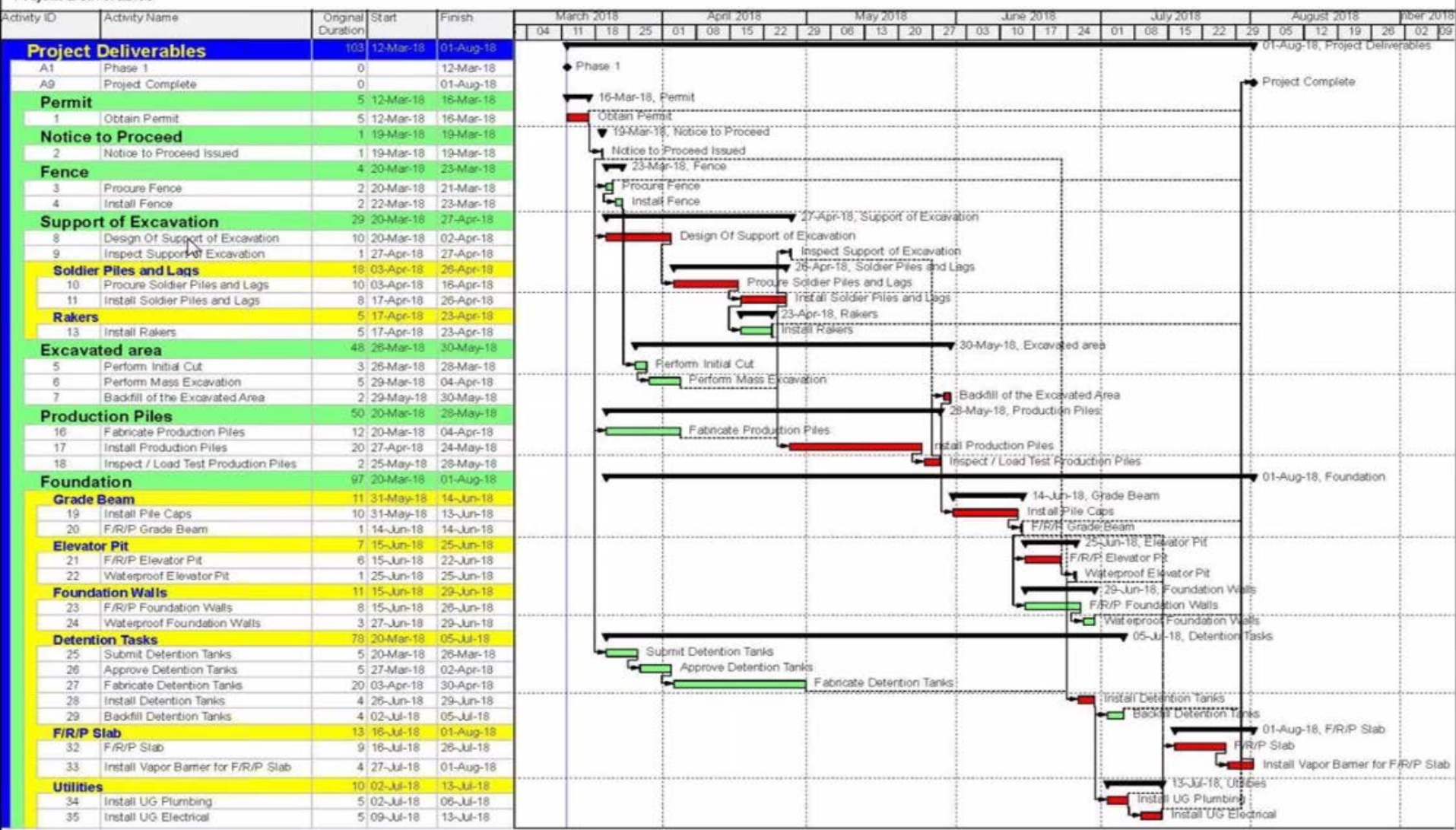


فرآیند یکپارچه برنامه ریزی، زمانبندی، ارزیابی و کنترل پروژه
استاندارد اجرایی: شناسایی و تعریف همه فعالیتها

WBS integration matrix – Case study: Improved schedule (Example)

□ Deliverable oriented view

Project Deliverables

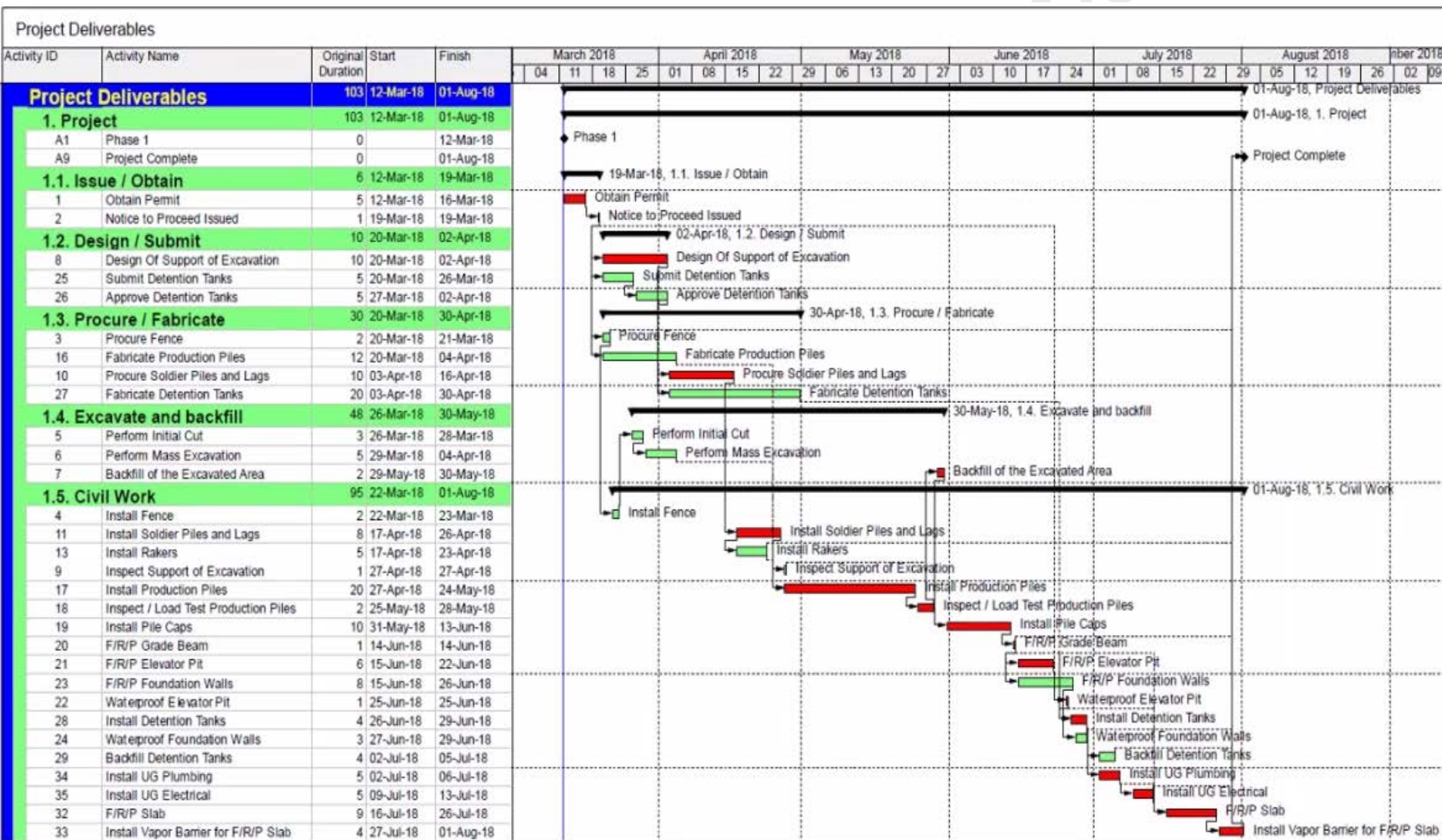




فرآیند یکپارچه برنامه ریزی، زمانبندی، ارزیابی و کنترل پروژه
استاندارد اجرایی: شناسایی و تعریف همه فعالیتها

WBS integration matrix – Case study: Improved schedule (Example)

□ Process oriented view



WBS integration matrix – CSI WBS (Example)

Relationship Between UNIFORMAT & MASTEFORMAT

Design Uniformat	Level 2	
	Level 3	Level 2
01 General Requirements	011 Standard Foundations	01 General Requirements
02 Site work	012 Spec. Foundation Cond.	02 Substructure
03 Concrete	021 Slab on Grade	
04 Masonry	022 Basement Excavation	
05 Metals	023 Basement Wall	
06 Wood & Plastic	031 Floor Construction	03 Superstructure
07 Therm. & Moisture Protect.	032 Roof Construction	
08 Doors & Windows	033 Stair Construction	
09 Finishes	041 Exterior Walls	04 Ext. Closure
10 Specialties	042 Ext. Doors & Windows	
11 Equipment	061 Partitions	05 Roofing
12 Furnishings	062 Interior Finishes	06 Int. Closure
13 Special Construction	063 Specialties	
14 Conveying Systems	081 Plumbing	07 Conveying Sys.
15 Mechanical	082 H.V.A.C.	08 Mechanical
16 Electrical	083 Fire Protection	
	084 Spec. Mechanical Syst.	
	091 Services & Distribution	09 Electrical
	092 Lighting & Power	
	093 Spec. Electrical Syst.	
	111 fixed & Movable Equip.	10 Gen. Cond. OH & P
	112 Furnishings	11 Equipment
	113 Special construction	
	121 Site Preparation	12 Site Work
	122 Site Improvements	
	123 Site Utilities	
	124 Off-Site Work	

WBS integration matrix – CSI WBS

- ❑ UniFormat is a **systems-based organization** of building content.
- ❑ MasterFormat is a **material-based organization** of building content.
 - ❖ In MasterFormat, a value of 03-31-00 is reserved for **Concrete Material**. All use of concrete will be specified with this **value of 03-31-00**.
 - ❖ However, UniFormat can be used to **differentiate** the **many places** concrete is used in a building.
 - ✓ *Examples of **UniFormat** codes that can identify how concrete is being used in a project are as follows:*
 - o A1010 Foundations
 - o A1020 Special Foundations (Caissons and Drilled Piers)
 - o A1030 Slab on Grade
 - o A2020 Foundations
 - o B1010 Structure
 - o C2010 Stairs
 - o G2030 Site Improvements

WBS integration matrix – Comparison with common approaches

Common Approach	Proposed Approach
There can only be one WBS per project.	There can only be one deliverable-oriented WBS and one process-oriented WBS per project.
WBS consists of a mixture of key deliverable, phases, or work packages.	A deliverable-oriented WBS consists of deliverables and a process-oriented WBS consists of phases and process steps.
WBS is used to define the project scope, based on which activities are identified using some form of a manual process.	WBS is used to systematically identify project activities using an integration matrix.
Process steps are not usually developed and documented in the form of a breakdown structure.	Process-oriented WBS should be treated as an essential input to the planning and project reporting processes.

Activity names

- ❑ A **consistent convention** for naming work activities should be **established early**.
- ❑ Names for **all activities**, including summary, milestone, and detailed activities, should be unique and as descriptive as necessary.
 - ❖ **Activity names** should be phrased in the present tense with verb-noun combinations, such as “**review basis of estimate**,” “**test level 4 equipment**,” or “**install and test workstations, room 714**.”
 - ❖ **Milestone descriptive names** should be **related** to an event or a deliverable, such as “Milestone B decision” or “level 4 test results report delivered.”

*In all cases, **descriptive names** should be unambiguous and should **identify** their associated product without the need to **review** high-level summary activity or preceding activity names.*



Activity names

1- Use an action word.

- ☐ (For example: Install Columns Formwork).

2- Don't abbreviate unless you absolutely have to.

- ☐ (For Example: "Inst frm wrk" BUT "Install Form Work")

3-Be descriptive

- ☐ (For example: Install Granite for North elevation).

4-Uniformity - Use the <Action Word> – <Deliverable> – <System> template. (Award Purchase Order (PO #) – Horizontal Pump)

5- Don't scream!

- ☐ THERE IS NO NEED FOR ALL OF YOUR ACTIVITIES DESCRIPTIONS TO BE CAPITALIZED!

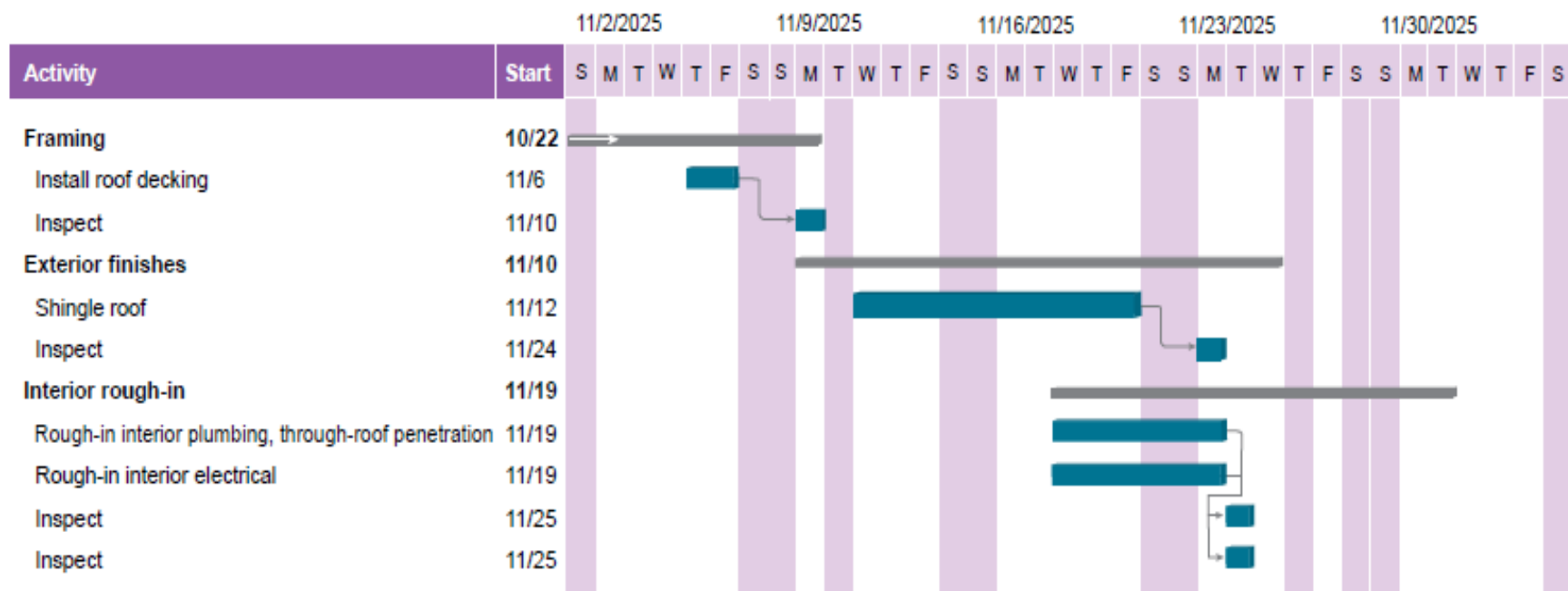
Activity names

- ☐ **Filtering** the schedule data to display **only critical path activities** will **not be helpful** if **activity names are redundant**.
- ☐ **Communicating** to **management** is **not useful** unless management **knows** which activity is meant.
- ☐ **Unique activity names** are **essential** if a schedule is to **produce reliable information** at the summary, intermediate, and detailed levels.
- ☐ Finally, **abbreviations** specific to **project** should be **minimal** and, if **used**, should be **defined** in either the WBS dictionary.

Activity names – Example

- ❑ It is **not clear which products** the **detailed inspection activities** are associated with **if** their **respective summary activities** and **preceding activities** are **not included**.
- ❑ **Repetitive naming** of activities that are **not associated** with specific products or phases makes **communication difficult** between teams.

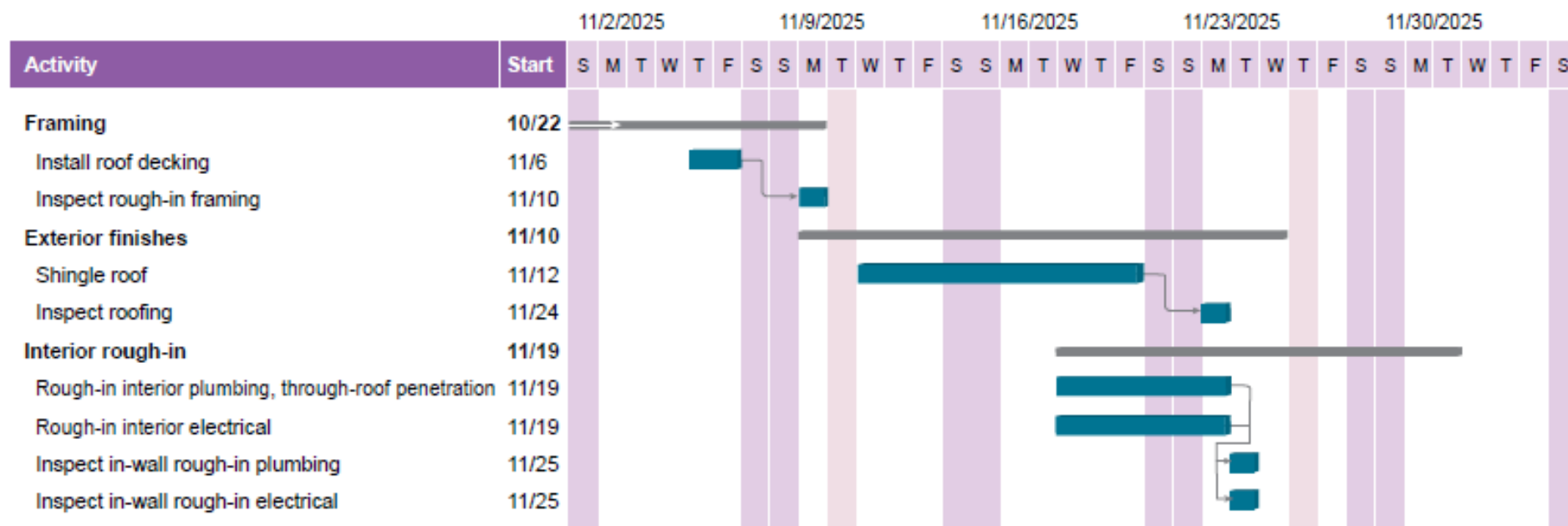
Redundant Activity Names



Activity names – Example

- ❑ Individual activities in **this example** are **easier to identify**.
- ❑ Activity names include their **respective products**, which are also identified for clarity in their summary activity.

Unique Activity Names



Source: GAO. | GAO-16-89G

Activity names – Example

- ❖ **“PL11-Planning Reports”**- does **not describe the action**.
 - ✓ Better example would be **“Develop PL11-Plan for producing CDRL A008 reports”**
- ❖ **“Training Clients”**- does **not relate to specific training** and can be confused with other training referenced without specific descriptions.
 - ✓ Better example would be **“Train Clients with TRN-3400 courseware”**

Activity codes

☐ Activities should be **consistent** with such information as the related:

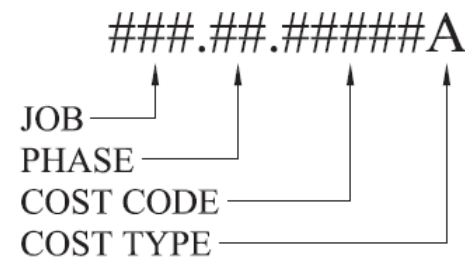
- ❖ WBS element
- ❖ Contractor
- ❖ Location
- ❖ Phase
- ❖ Contract line item number (CLIN)
- ❖ Work package number
- ❖ Control account
- ❖ SOW paragraph as applicable
- ❖ Job cost ledger
- ❖ Equipment ledger

Activity codes based on coding system presented for Job Cost Ledger

❑ The job cost code of **102.01.07200L**

❑ **All parties**—estimators, field employees, the accounting department, and management— **must be using the same coding.**

❖ If **field employees** code framing hardware to a **different code** than the **estimators**, tracking the project's costs against the **estimator's budget**.....



Sample Job Cost Code

Activity codes

- ❑ Each activity is better to be associated with a **unique alpha-numeric code**.
- ❑ This **unique code** should preserve the **hierarchical relationship** between the **activity** and its **parent elements**.

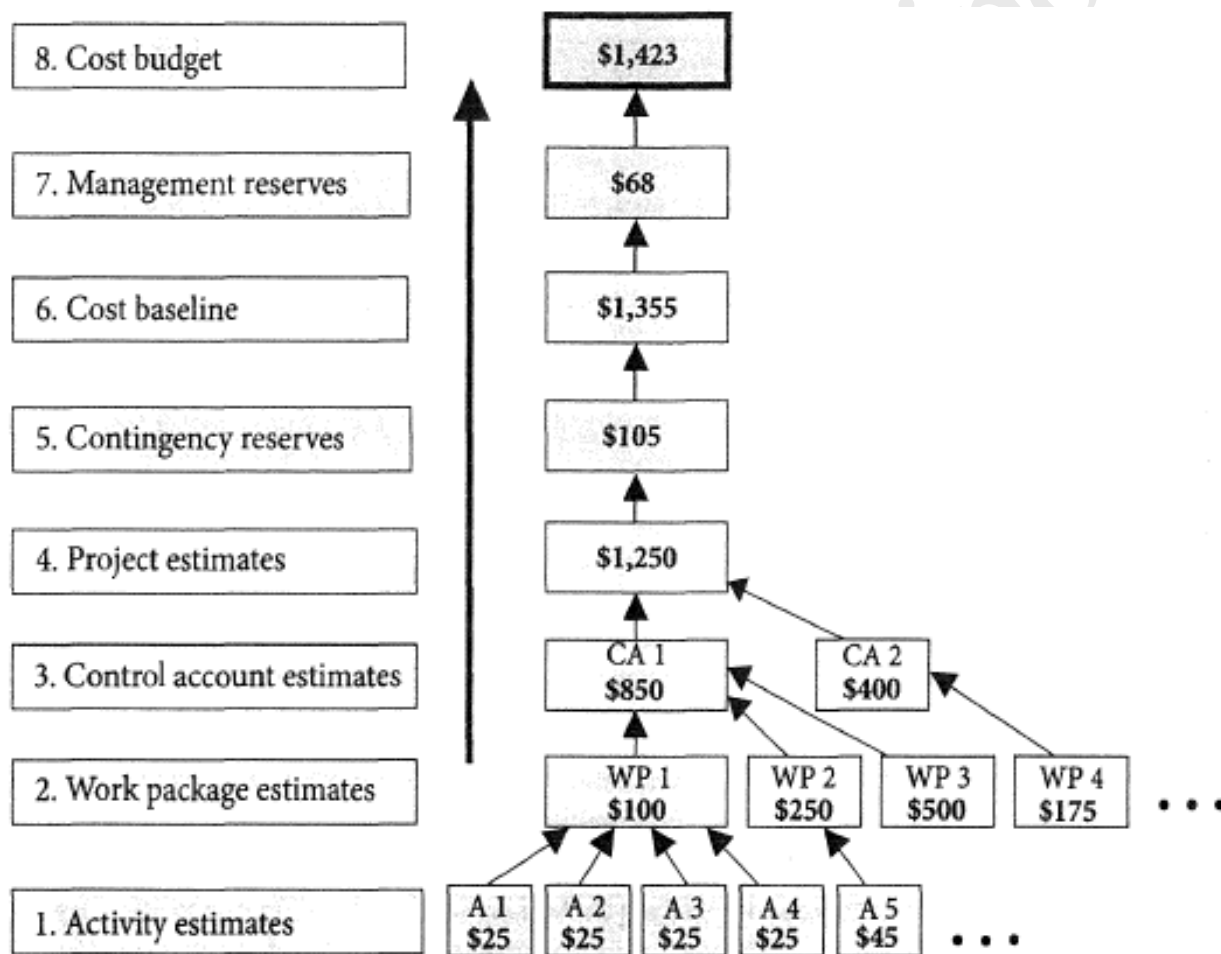
For example, the activity “Test software Unit 1A” could be mapped to the code “ADZT125” to represent the Alpha project (A), development phase (D), Zeta prime contractor (Z), testing activity (T), and activity ID number (125).

- ❖ You can also use the IMP format as mentioned previously.

Control account (CA)

- ❑ The Control Account (CA) is a **management control point** where scope, budget, actual cost, and schedule are integrated to facilitate effective program management.
- ❑ Control accounts are the **point of the WBS** where **assignment of work responsibility** is given to a Control Account Manager (CAM).
- ❑ Each discrete IMS task/activity with resources is assigned and traceable to **only one Control Account**.

Control account (CA) – Example



Organizational breakdown structure (OBS)

❑ OBS (**Organizational Breakdown Structure**) is used to define the responsibilities for the activities.

❖ Every activity should be **identified** with only one activity owner.

✓ The activity owner is the **person responsible** for:

- Understanding the **scope of work** and **objectives**
- Defining the **entrance** and **exit criteria**
- Providing the **schedule updates** to the **scheduler**
- Understanding any **variances** that might occur.

Each IMS activity/task is assigned and traceable to one Control Account within the OBS.

WBS dictionary – Example

❑ WBS must have an associated WBS dictionary that clearly defines

❖ Scope of each **individual WBS element**

❖ Scope of related **schedule activities**

WBS Level	WBS Code	WBS Name	WBS Description	PWS/SOW Mapping
1	1	Aircraft System	X Series Aircraft System to fly to the moon	
2	1.1	Air Vehicle	X Series Air Vehicle to fly to the moon	
3	1.1.1	Air Frame	X seriease air frame	
3	1.1.2	Avionics	Brains behind the air frame	
4	1.1.2.1	Comms	Communications	
5	1.1.2.1.1	Antenna	Thinging to pick up signals	1.1, 1.2, 1.3.1, 1.13.1
5	1.1.2.1.2	Receiver	Box to interpert signals picked up by Antenna	1.1, 1.2, 1.13.2
5	1.1.2.1.3	Transmitter	Box to send out signals when we talk	1.1, 1.2, 1.3.1, 1.13.3
4	1.1.2.2	Navigation	Back seat driver	
4	1.1.2.3	Fire Control	Off/Def weapons in case we run into aliens	
3	1.1.3	Propulsion	Engine to propel x series air frame	
2	1.2	Training	Training for operation and maintenance of X Series Air Vehicle	

WBS dictionary – Example

No.	Item	Description
1	WBS Code	Enter the WBS Title and identify it as a Work Package, Planning and/or Control Account.
2	Responsible Organization/ Individual	Name the single organization, group or individual that has been assigned sole responsibility for making sure the Work Package is completed. Include contact information.
3	Description	Define Work Package boundaries. Frame the scope content so that it is clear what scope is included and excluded. If it is a Planning Package, describe the known and unknown scope.
4	Deliverables	Identify the product, service or results created when all of the work in this Work Package is complete. Include any critical intermediate deliverables.
5	Acceptance Criteria	Describe the functional and physical requirements in order to meet customer expectations and quality requirements. Include any unique approvals required for acceptance.
6	Budget	Designate the budget for this Work Package, plus any critical resource information and assumptions.
7	Milestones	List any Start Dates, End Dates, Intermediate Milestones, Interdependencies, Constraints and any assumption for the deliverables.
8	Risks	Include any known threats and opportunities with response strategies.
9	Additional Information	Describe any additional information, such as references, related work packages, etc.
10	Approvals:	Date: Rev:

IMPORTANT NOTE

موضوعات تکمیلی بیشتری از آموزش برای این بخش در سایت و برای اعضای کانون
قرار خواهد گرفت (احتمالا بخش TV)

1. WBS
2. Activity ID Coding
3. Activity Coding
4. Milestone
5. Activities

Checklist For a Scheduler For Capturing All Activities

- ☐ A WBS elements are linked to one another with logical relationships and lead to the end product or final delivery.
- ☐ The schedule clearly reflects the WBS and defines the activities necessary to produce and deliver each product.
- ☐ The schedule reflects all effort (steps, events, work required, and outcomes) to accomplish the deliverables described in the program's work breakdown structure.
- ☐ The IMS includes the summary and intermediate and all detailed schedules.
- ☐ The owner-owned detailed schedule includes all activities the owner, its contractors, and others must perform to complete the work.
- ☐ The schedule contains primarily detail activities, and milestones are not used to represent work.

- ☐ If the owner's project management office and its contractor use different scheduling software packages, a process is defined to preserve integrity between the different schedule formats, and the converted data are verified and validated when the schedules are updated.
- ☐ Level-of-effort (LOE) activities represent effort that has no measurable output and cannot be associated with a physical product or defined deliverable.
- ☐ Activity names contain noun-verb combinations, are descriptive, and are clear enough to identify their associated product without the need to review high-level summary or predecessor activity names.
- ☐ Activities within the schedule are easily traced to key documents and other information through activity or task codes.

An Auditor's Key Questions and Documents for Capturing All Activities

Key questions

1. Is there an IMS for managing the entire program (not just a block, increment, or prime contractor)? Is the schedule defined at an appropriate level to ensure effective management?
2. Is the IMS maintained in scheduling software and linked to external, detailed project schedules?
3. How does management ensure the accuracy of reported schedule information? Do the government program management office and contractors have different scheduling software systems? If so, how is integrity preserved and verified when converting the schedule?
4. Does the IMS include government, contractor, and applicable subcontractor effort?

Key questions

5. Does the schedule reflect the program WBS and does the WBS allow tracking key deliverables? Does every activity trace to an appropriate WBS element, and do the activities define how the deliverables will be produced? Does the schedule WBS map to the cost estimate WBS? Is there a WBS dictionary?
6. Are key milestones identified and are they consistent with the contract dates and other key dates management established in the baseline schedule?
7. Does the schedule have clear start and finish milestones? Are there too many milestones in relation to detail activities?
8. Are activities within the schedule easily traced to key documents and other information through activity or task codes? Are all contractor activities mapped to the contract statement of work (SOW) to ensure that all effort is accounted for in the schedule?

Key questions

9. Are activity names unique and descriptive? Are activities phrased in verb-noun combinations (for example, “develop documentation”)? Are milestones named with verb-noun or noun-verb combinations (for example, “start project” or “project finished”)?
10. Are level-of-effort activities clearly marked?
11. Does the schedule include significant risk mitigation efforts as discrete activities? If not, how are they documented and tracked?

Key Documents

1. Work breakdown structure (WBS) and dictionary
2. Statement of work (SOW), integrated master plan (IMP) and mission requirements, as applicable
3. SOW crosswalk to the WBS and schedule activities, as applicable
4. Contractor WBS to program WBS crosswalk
5. Schedule custom fields and activity codes dictionary and LOE field identification
6. Activity codes used to organize and filter the activities into categories as necessary to confirm a complete scope of work
7. Plans and documentation used for defining activities, such as the systems engineering plan, software development plan, risk management plan, and master test plan.



فرآیند یکپارچه برنامه ریزی، زمانبندی و کنترل پروژه

بخش چهارم:

استاندارد اجرایی: اولویت بندی و تعیین روابط بین همه فعالیتها

ارائه دهنده:

سید محمدرضا علوی پور

(PhD, PMP, CMIT)

What we will learn?

☐ Introduction

☐ Constraints

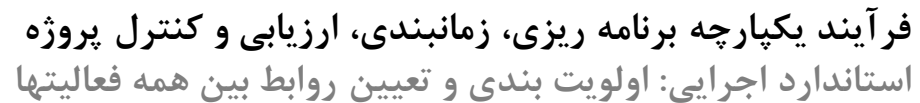
- ❖ Introduction
- ❖ Types of constraints
- ❖ Types of logics
- ❖ Date constraints
- ❖ Important points

☐ Sequencing all activities

- ❖ Development of activity sequence
- ❖ Predecessor and successor logic
- ❖ Summary logic
- ❖ Incomplete and dangling logic
- ❖ Lags
- ❖ Leads
- ❖ Path convergence

What we will learn?

- ☐ Checklist for a scheduler for sequencing all activities
- ☐ An auditor's key questions and documents for sequencing all activities
 - ❖ Key questions
 - ❖ Key documents



- ❑ This is a cyclic process
- ❑ Elaborated through rolling wave process.

A process for creating and maintaining reliable schedules

Process step	Corresponding action	Section
Capture all activities	Capturing all activities	3
Create logically sequenced activity network	Sequencing all activities	4
Estimate work and durations and assign resources	Allocating and leveling resources	5
	Determining and assigning durations to activities	6
Validate critical path and reasonable total float	Critical path determination, assessment, and validation	7
	Ensuring reasonable total float	8
Analyze schedule risk	Schedule constructability and risk analysis	9
Verify and validate traceability	Verifying that the schedule can be traced horizontally and vertically	10
Set and document baseline	Setting, monitoring, controlling, and Maintaining a baseline schedule	11
Update, revise, and manage change	Setting, monitoring, controlling, and Maintaining a baseline schedule	11
	Updating the schedule using actual progress and logic	12

Introduction

The AACE International, in *Recommended Practice No. 24R-03, "Developing Activity Logic,"* defines relationships as:

a recommended practice of AACE International, development of activity logic (also called network logic) in planning and scheduling provides guidelines for the sequencing of activities in a logical way generally before duration estimating can be performed. Logic is generally determined before durations are considered. Logic development methods include precedence diagramming, or arrow diagramming methods. Logic is the set of activities and dependency relationships between them. Logic dictates the planned sequencing of activities. A network diagram is often used to illustrate the logic. (AACE, 2004, p. 2)

Introduction

- ❑ The **sequencing of activities** in a schedule is **based** on the **occurrence of activities relative to other activities**.
 - ❖ This is simply a **logical progression** of the **anticipated occurrence** of each activity relative to other activities.
- ❑ By establishing the **network logic**, the schedule can **predict**:
 - ❖ Effect on the project's planned finish date
 - ❖ Misallocated resources
 - ❖ Delayed activities
 - ❖ External events
 - ❖ Scope changes
 - ❖ Unrealistic deadlines
 - ❖ Effect of risk events

Introduction

- ❑ Ordering activities is **not necessarily** as straightforward as it may seem.
- ❑ The reason why **two activities** must be done in a **particular order** can be termed a “constraint.”
- ❑ **Without constraints** on a project, **all activities** can **theoretically begin** on the **first day** of construction.
 - ❖ Constraints must be considered in order for a network to be useful.
- ❑ The **duration of a project** is **contractually constrained** by the length of time specified in the **contract documents**.

**Re-sequencing may be required once durations are applied,
typically to meet an imposed constraint.**

There are also a number of other types of constraints that might impact project duration.

Constraints

Introduction

❑ Constraints may be **imposed** on:

1. Individual activities
2. Group of activities
3. Entire project

Introduction

❑ Constraints should be **minimized** because **excessive constraints** can create **problem**

in network logic. Why?

1. **Reduce** scheduling **flexibility**
2. **Lengthen** project **duration**
3. **Increase** project **cost**
4. **Confuse** basic **scheduling logic**

Types of constraints

1. Physical constraints
2. Resource constraints
3. Safety constraints
4. Financial constraints
5. Environmental constraints
6. Management constraints
7. Contractual constraints
8. Productivity constraints
9. Regulatory constraints

Types of constraints – How to handle?

1. Once the basic **network** is **completed**, the constraints should be **imposed** in some **sequence**.
 - ❖ This will **generally** be **done** with scheduling software, the **impact** of particular constraints can be **readily evaluated**.
2. When the impact of a **particular constraint** is especially **severe**, the **scheduler** will probably reassess the **need** for the **particular constraint**.
 - ❖ In many instances, a constraint can **change** without seriously **compromising** **construction progress**.

Types of constraints – How to handle?

❑ Only physical constraints should be entered during the early development of scheduling.

❖ Others should be **deferred** until the **actual scheduling** of activities to:

1. See whether the **constraints** are **met** by the **basic schedule**
2. If not, can it be addressed by the shifting of activities within their available float?

Any constraints **not handled** in this manner can be **addressed** in **another iteration** of the network development process.

How to categorize constraints?

☐ Constraints categorized into:

1. Logic

- i. Contractual logic
- ii. Mandatory (physical logic)
- iii. Discretionary logic

2. Date constraints

Types of logics

❑ Schedule's sequencing of the work activities

1. Contractual logic
2. Mandatory logic
3. Discretionary logic
 - ✓ Contractors choice
 - ✓ Most likely to be out-of sequence

Types of logics – Contractual logic

- ❑ This logic is **derived** from the **contract itself** and is **usually** identified by the **owner**
- ❑ It **may mandate** one section of the project **being completed first**, or some **other milestone** being **accomplished prior** to **other work** being undertaken

➡ *For example, a **school project** may **mandate** that the **classrooms** be **opened** by the start of school in **September**, but **allow** the **gymnasium** facilities to **lag until** the start of winter*

There **may be no logic** behind this, but the **contract mandates** that

Types of logics – Mandatory logic

- ❑ This **logic** relationship is **common** on **projects**

*For example, This is the logic that **requires** the **excavation** to **precede** the **footings**, followed by the **foundation**, followed by the **structure**, and so on*

- ❑ It is **occasionally** possible to **cheat** the **requirements of physics** – **for example**, **constructing** the **building** from the **top down**.

- ❑ **Most construction schedules** are developed **around** the **contractual** and **mandatory** logic requirements

Types of logics – Discretionary logic

- ❑ This is the logic that is **developed when**, for the **particular project** in question, there is **no contractual** or **physical necessity** to perform the work in a certain order
 - ❖ At the **same time**, discretionary logic **recognizes** that there are **often valid reasons** for performing the work in a **certain order**
- ❑ Happens when there is a **choice**

*For example, if the contractor is erecting the partitions and sheetrock on the **fifth floor** of a **building under construction**, there is probably **no contractual** or **mandatory logic** for erecting it from **south to north** rather than **north to south**.*

Date constraints

- ❑ Constraints can be placed on an **activity's start** or **finish date** and can **limit** the **movement** of an activity.
- ❑ Date constraints can be placed in **two ways**:
 1. Soft constraints
 2. Hard constraints
- ❑ **Both** can be applied to the **start** or the **finish** of an activity.
- ❑ The **purpose** of a constraint is to **limit when** an activity can **start** or **finish**.
- ❑ **Milestones, crucial material deliveries, and subcontractor scheduling** can create the **need** for **constrained activity** starts and finishes.

*Misuse of **date constraints** in the IMS may have unintended consequences including distortions to float and critical path.*

Date constraints – Soft constraints

1. Start no earlier than (SNET) or scheduled early start (SES)

$$ES \text{ (constrained activity)} = \text{MAX (Calculated ES or SES)}$$

2. Finish no earlier than (FNET)

$$LF \text{ (constrained activity)} = \text{MAX (Calculated LF or FNET)}$$

Date constraints – Hard constraints

1. Start no later than (SNLT)

$$ES (\text{constrained activity}) = \text{MIN} (\text{Calculated ES or SNLT})$$

2. Finish no later than (FNLT) or Scheduled late finish (SLF)

$$LF (\text{constrained activity}) = \text{MIN} (\text{Calculated LF or SLF})$$

Date constraints – Hard constraints

- ☐ These types of constraints **prevent** activities from **starting** or **finishing** on **any day other than** the **date assigned**.
- ☐ **When** an activity must start or finish on a **specific date**
- ☐ **Operated differently** than **early** and **late start** and **finish** times.
- ☐ Two types:
 1. **Must start on (MSON)**
 - ❖ It **prevents** an activity from **starting** any **earlier or later** than a certain date
 2. **Must finish on (MFON)**
 - ❖ It **prevents** an activity from **finishing** any **earlier or later** than a certain date

Date constraints – Summary

- ❑ Start no earlier than (SNET) and finish no earlier than (FNET) constraints **delay** activity starts and finishes **even if** predecessor durations allow them to **occur earlier**.
 - ❖ These are soft constraints.
 - ❖ Their **use** must be justified because, in general, **project management's wanting** to **not start or finish** an activity **as soon as possible** may be questionable.

Do not use a lag in place of a soft constraint for date-dependent activities

Date constraints – Summary

- ❑ Start no later than (SNLT) and finish no later than (FNLT) constraints **prevent** activities from **slipping**. Therefore,
 - ❖ These are hard constraints.
 - ❖ Their **use** is discouraged.
 - ❖ They should never appear in the **schedule baseline**.
 - ❖ If they are not properly justified in **working** schedules, they must be **immediately questioned**.

Constraints have the potential to create negative float

Date constraints – Summary

- ❑ Must start on (MSON) and must finish on (MFON) constraints **prevent** activities not only from **slipping** but also from **accelerating**. Therefore,
 - ❖ These are hard constraints.
 - ❖ Their **use** is **discouraged**.
 - ❖ They should **never appear** in the **schedule baseline**.
 - ❖ If **not properly justified** in **working schedules**, they must be **immediately questioned**.

Date constraints – Important points

- ❑ **Constraints** are **restrictions** set on the start or finish dates of tasks.
 - ❖ These restrictions
 1. **Place conditions** on **how dates** are calculated
 2. **Affect total float calculations** within the schedule
 - As a result, **dates** and **total float values** may **calculate differently** when performing the **forward pass and/or backward pass** in a logic network schedule.
- ❑ The **forward pass** calculates the **Early Start (ES)** and **Early Finish (EF)** dates for all milestones and activities.
- ❑ The **backward pass** calculates the **Late Start (LS)** and **Late Finish (LF)** dates for all milestones and activities.

Applying constraints to tasks and milestones impacts these network calculations

Date constraints – Important points

- ❑ Many activities with **date constraints** are typically **substitutes** for **logic** and can signify that the schedule
 1. Was **not well planned**
 2. May **not be feasible**

Constraints can cause *misleading risk analysis* results because they *override network logic*.

What other references have said?

Date constraints – Important points

PMI Best practice

Default Condition

Preferably, a project should only have two constraints within the schedule: the project start and project finish dates. All other activities should be driven by logic. The use of constraints in a schedule alters intent and logic unless specifically outlined by narrative as to the nature and intent of the constraint.

Date constraints – Important points

DCMA 14 points assessment

Hard Constraints

Under 'Constraints' in a project schedule are identified Hard Constraints (Mandatory Start and Mandatory Finish) and Soft Constraints (As Late As Possible, Start On, Start On or Before, Start On or After, Finish On, Finish On or Before and Finish On or After).

DCMA recommends 5% of the constraints used, be Hard Constraints because they can provoke problems in the schedule. Hard Constraints can prevent the schedule to be logic-driven and create negative lags. In other words, Hard Constraints should be avoided since they can obstruct the logic of flowing naturally. Soft Constraints, on the other hand, allow the schedule to be logic-driven and they are not critical to the schedule.

So, what should be done?

Date constraints – Important points

PMI Best practice

Recommended Practice

It is difficult not to use some constraints in a schedule, so use them sparingly and correctly. Limit the number of constraints used within a schedule, only using constraints on dates that drive the success of the project. This will allow one to see the negative float faster and take corrective actions early.

Constraints are generally necessary to represent logic external to the project. Contractual dates represented by milestones are a good example of this kind of external logic, as are delivery dates which may also represent contractual agreements. One may find it helpful to place constraints only on milestones and use each milestone to describe the external relationship represented.

Constraints should only be used (and must be used) where the “real world” interferes with schedule logic and you cannot achieve the preferred dates through relationships and durations.

During the development of the schedule, identify the constraints used, make a copy of the schedule, and then run it without constraints to see how the constraints affect the schedule. Constraints should be used as a monitoring tool, not a scheduling tool to benchmark critical milestones within the project. Minimize the use of constraints and let logic drive the schedule, which is the basic premise of CPM scheduling.

Date constraints – Important points

Constraints vary from software package to software package, so the scheduler must be aware of all the features within the scheduling software packages, as the constraint has the most power to distort a schedule.

It is one of the most misused features within schedules today.

Date constraints – Important points

❑ Do not assume that **all scheduling tools** handle 'constraints' in the **same manner**

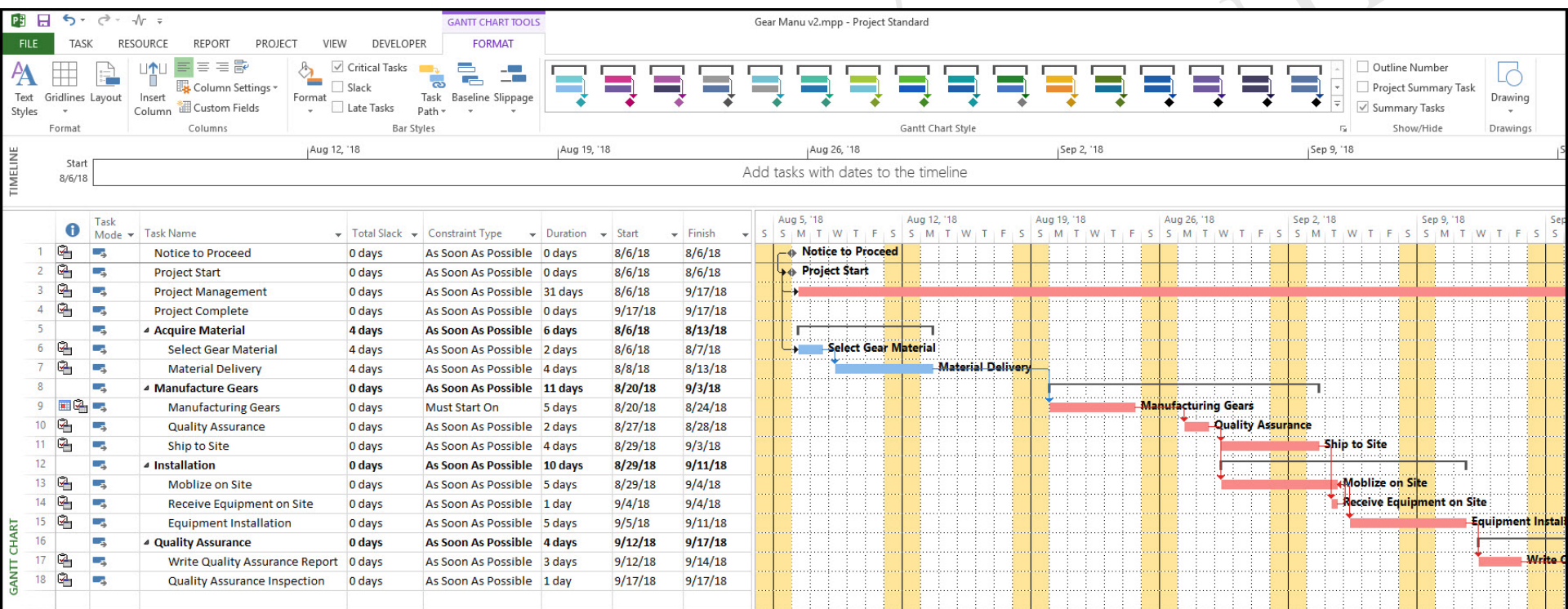
❖ For example in MSP:

- ✓ As soon as possible (ASAP)
- ✓ As late as possible (ALAP)

Date constraints – Important points

✓ As soon as possible (ASAP)

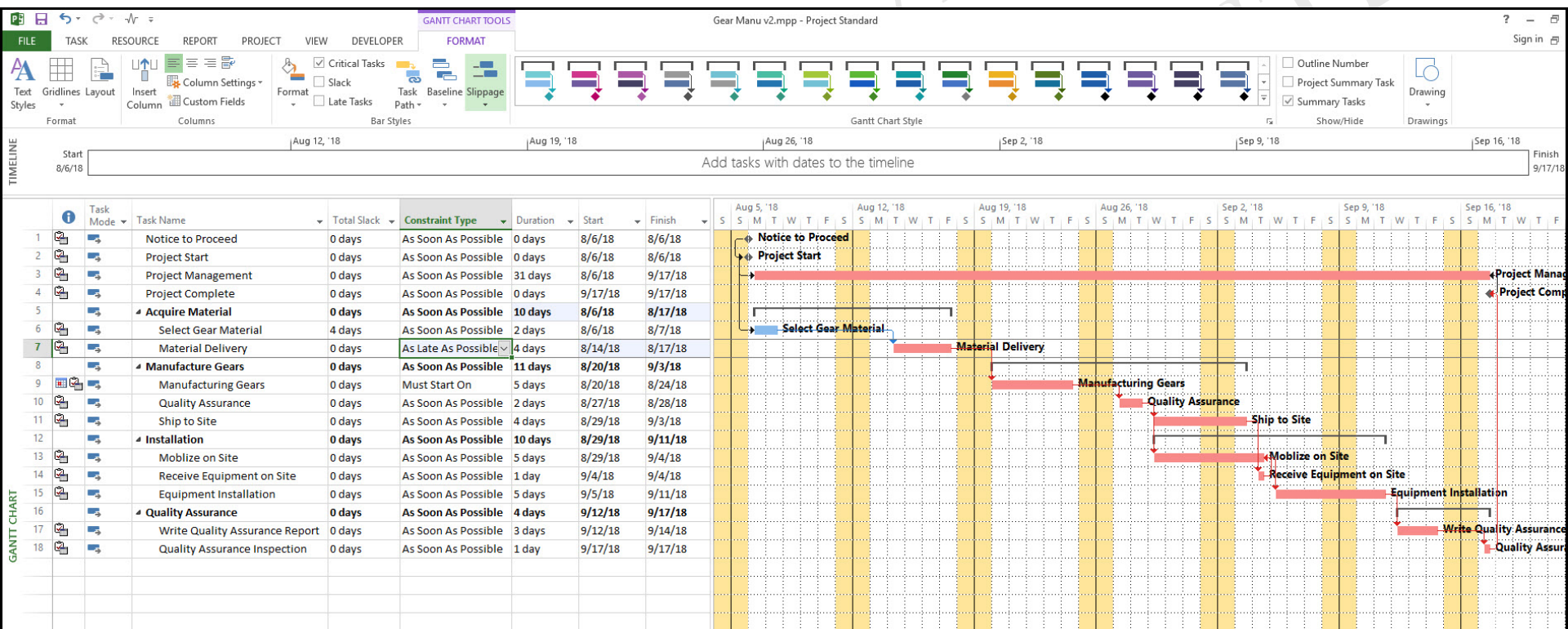
- If you want to use lean approach (JIT) and deliver material with commencement of manufacturing, this Material Delivery with **ASAP** constraint avoids that.



Date constraints – Important points

✓ As late as possible (ALAP)

- With this constraint **total float has been removed** and material delivery occurs on **Friday evening**.



Date constraints – Important points

✓ As soon as possible (ASAP)

- Default constraints for tasks in MSP
- This is ideal for most forward scheduling situation

✓ As late as possible (ALAP)

- Not recommended for general use, as it burns total float to align to the completion of predecessor task with the commencement of its constrained successor.
 - ❖ However, in some tools, **ALAP** will **only remove the free float, not the total float.**

Both ASAP and ALAP are soft constraints

Date constraints – Important points

موضوعات تکمیلی بیشتری از آموزش برای این بخش در سایت و برای اعضای کانون
قرار خواهد گرفت (احتمالا بخش TV)

1. **Mandatory constraints**
2. **Early constraints**
3. **Late constraints**
4. **Other constraints**

Sequencing All Activities

Development of activity sequence

❑ For **each identified activity**, the following **must be determined**:

1. Which activities must **precede** it?
2. Which activities must **follow** it?
3. Which activities can be **concurrent** with it?

*All of above questions can be addressed by simply identifying, for **each activity**, all of the immediately preceding activities (IPAs).*

Development of activity sequence

- ❑ When the **IPAs for all of the activities** are described,
each activity's succeeding activities will have been generated,
along with the **activities** that can occur at the **same time**.

Once the IPA list is completed, the network can be created.

Development of activity sequence – Example

- ❑ **Sample activity** list and associated list of IPAs.
- ❑ **Procurement activities** (“Purchase steel reinforcement” and “Order concrete”) tend **not to have predecessors** because normally they can be done without anything preceding them.
- ❑ At this point **only the logic** of the process is being **defined**.

Sample Activity List with IPAs Project to Construct Concrete Footing		
Activity Label	Activity Description	IPAs
A	Lay Out Foundation	—
B	Dig Foundation	A
C	Place Formwork	B
D	Place Concrete	G, H
E	Purchase Steel Reinforcement	—
F	Cut and Bend Steel Reinforcement	E
G	Place Steel Reinforcement	C, F
H	Order Concrete	—

Sample Activity List with IPAs.

What has been done so far?

- ❑ Statement of work (SOW)
- ❑ Integrated master plan (IMP)
- ❑ Integrate master schedule (IMS)
 - Work breakdown structure (WBS)
 - Activities
 - Constraints
 - Types of logic related to constraints
 - Date constraints
 - Logic
 - Lag
 - Lead

- ❑ After schedule tasks are created, they are **“linked”** to show their **logical dependencies**.
- ❑ These **logical ties** in combination with the **task durations**, **date dependent constraints** and **lags** define the **anticipated** program **work flow** and are the foundation for the
 1. Schedule date calculations
 2. Critical path calculations
- ❑ All discrete tasks/milestones (**excluding LOE** and **Summary tasks**) should have **at least** **one predecessor** and **one successor**.

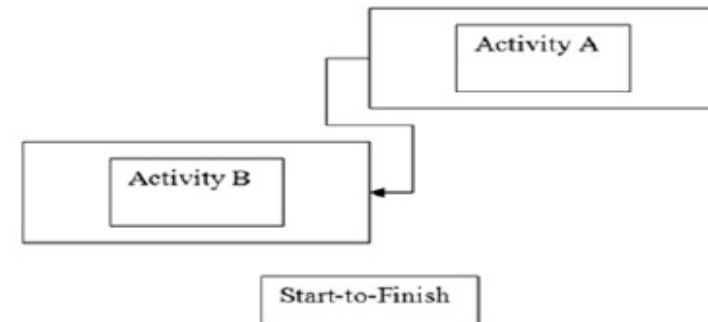
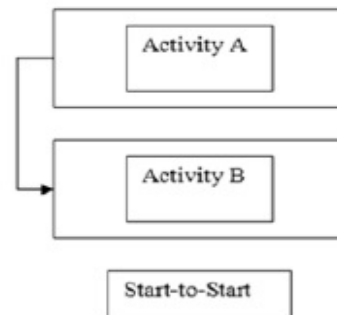
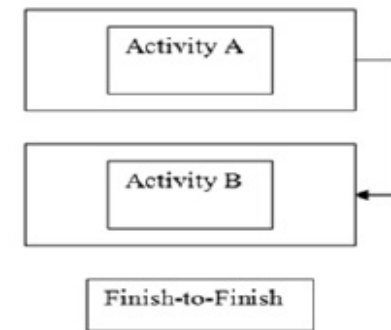
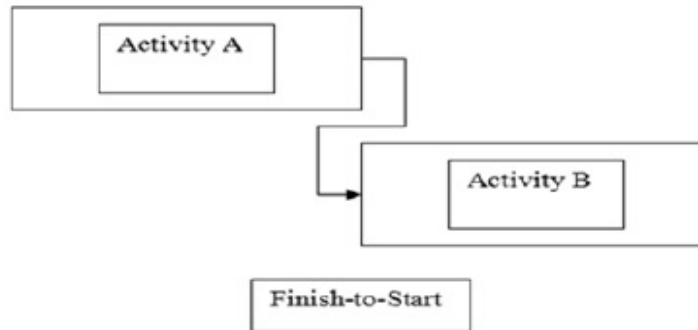
Logic – Predecessor and successor logic

❑ **Logic (relationships)** is what determines the sequence and order of activities in the **schedule network**.

❑ Logic relationship linking a predecessor and successor can take one of four forms:

1. Finish-to-start
2. Start-to-start
3. Finish-to-finish
4. Start to finish

One of them is very discouraged??



Logic – Predecessor and successor logic: Finish to start

- ❑ A **successor** activity **cannot** start until the **predecessor** activity **finishes**
- ❑ The **most straightforward** logical link between a predecessor and successor.
- ❑ The **default** in most scheduling programs.



Source: GAO. | GAO-16-89G

Logic – Predecessor and successor logic: Start to start

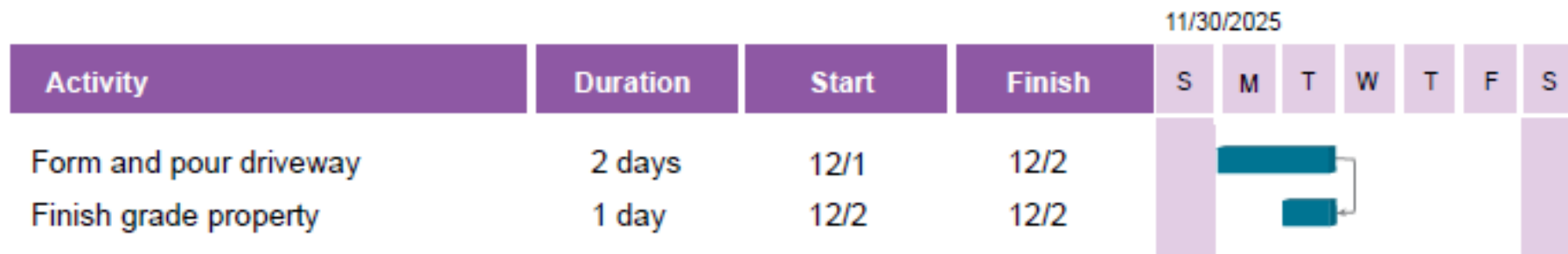
- ❑ A successor activity cannot start until the predecessor activity starts.



Source: GAO. | GAO-16-89G

Logic – Predecessor and successor logic: Finish to finish

- ❑ A successor activity cannot finish until the predecessor activity finishes.



Source: GAO. | GAO-16-89G

Logic – Predecessor and successor logic: Start to finish

- ❑ The start-to-finish (S-F) link is a **theoretical logical links**.
- ❑ It has the **bizarre effect** of **directing** a successor activity not to finish until its predecessor activity starts,
in effect **reversing the expected flow** of sequence logic.
- ❑ Its use is **widely discouraged** because it is **counterintuitive** and it **overcomplicates** schedule network logic.



*Examples of activity sequences used to justify the existence of an **S-F relationship** can usually be rewritten in simple **F-S** logic*

Logic – Predecessor and successor logic: Start to finish

❑ The first question is

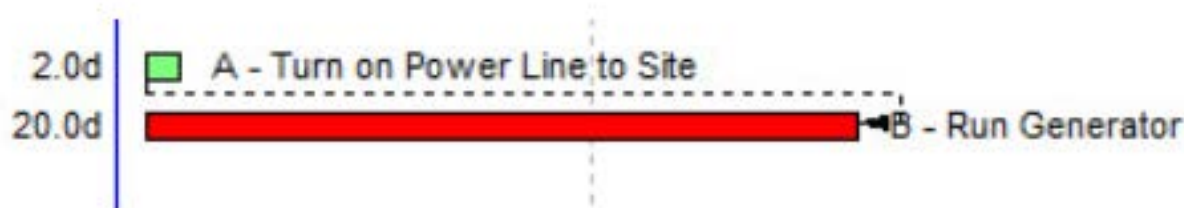
“Under what circumstances might one need to model a situation where something cannot finish until something else starts?”

❖ Dozens of schedulers have never found a really good actual live-use example where this relationship is being used.

Scheduling guidelines that actively discourage the use of SF relationships, largely because few consumers of schedule output truly understand what the scheduler is trying to model.

Logic – Predecessor and successor logic: Start to finish

- ❑ The best written example by Daniel L. Williams, PhD and Elaine Britt Krazer, PMP called Oracle Primavera P6 Version 8: Project and Portfolio Management.
- ❖ Imagine that you are **building a facility** in an area that has **no power transmission lines.**
 - ✓ Activity B can be: run a generator to power up site X.
 - ✓ Activity A can be: turning on a power line to the site, which is being built while the rest of the project progresses.
 - **B must continue** as long as **A has not started.**
 - It is **not necessary that B finish once A starts**, but it is **allowed to finish once A starts.**



Logic – Predecessor and successor logic: Start to finish

- ❑ Further confusion comes when **thinking** about SF linked activities in context with other preceding or succeeding activities.

*It can be hard to grasp what a Start to Finish link is buying us in terms of
how the critical path will be affected and
how these predecessors and successors will be impacted.*

Logic – Predecessor and successor logic: Start to finish

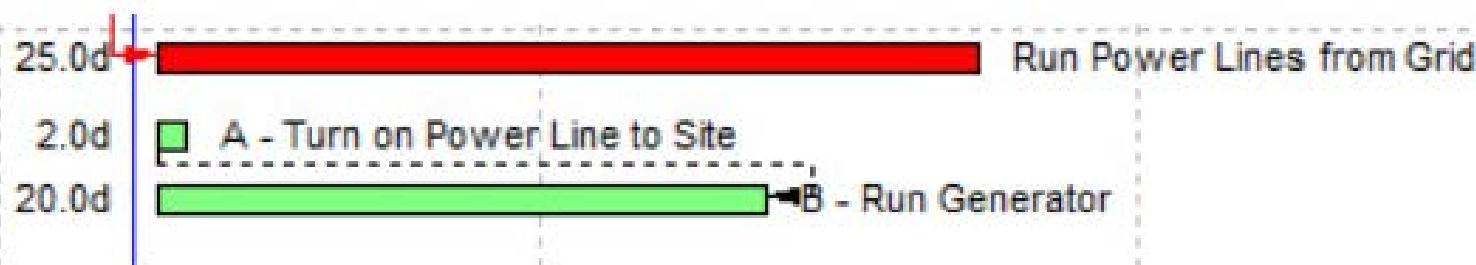
❑ In order to power up the lines,

there must be **some effort to get the cable run** from the power grid to the site.

❖ So this would **precede the Turn on Power Line** to Site activity.

❖ So let's say it will **take 25 days to run the cable** out to the site,

so we add 'Run Power Lines from Grid' activity to the scenario.



Logic – Predecessor and successor logic: Start to finish

- ❑ Let's now make the 'Run Power Lines' activity a **Finish to Start** predecessor of 'Turn on Power Lines' and watch what happens.



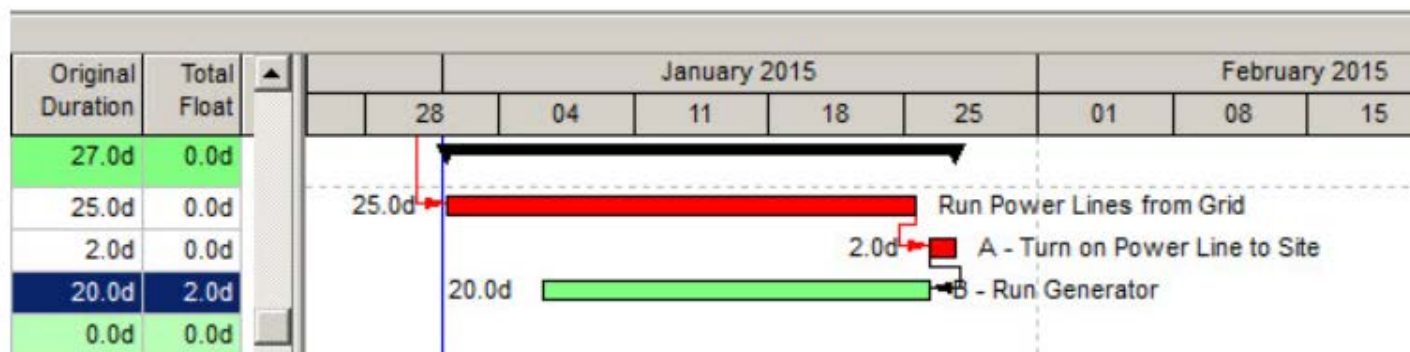
Logic – Predecessor and successor logic: Start to finish

Now what do we have?

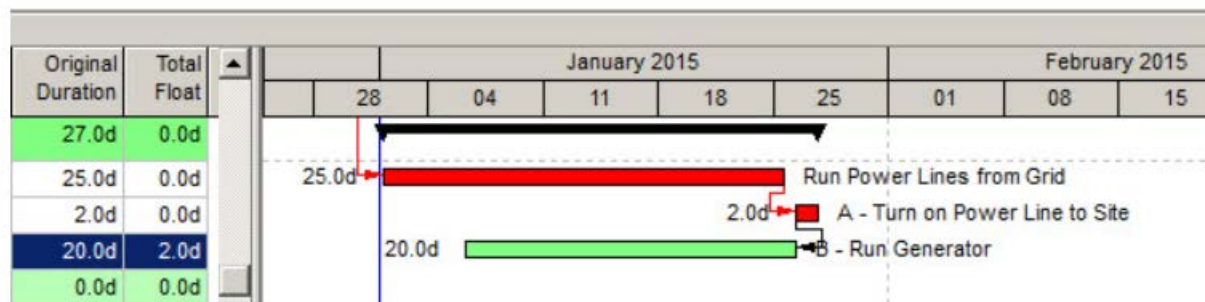
- ❑ That's interesting; the 'Turn on Power' activity has **dragged** the 'Run Generator' activity out the **extra 5 days**.
- ❑ What is also interesting is that if you look at the **Total Float** value for the **Run Generator activity**, it only **shows 2 days**.

Why only two days? Didn't it just slip 5 days further out from the current Data Date?

Think of it as delay analysis!



Logic – Predecessor and successor logic: Start to finish

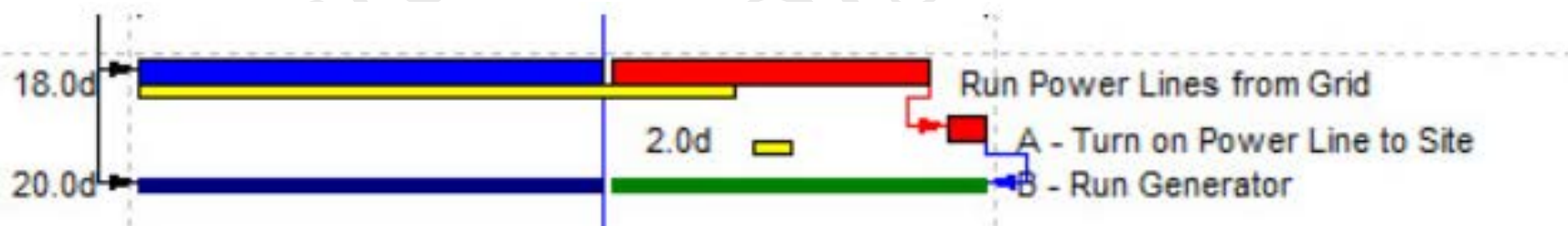


- ❑ Now you might be wondering **why B** is apparently showing a **2 day float on the late dates**,
 but a **3 day float on the early dates**

Logic – Predecessor and successor logic: Start to finish

Technically it is correct.

- ❑ But what it buys us, versus the headache we get trying to figure out how it affects the critical path is questionable.
- ❑ We could just **make Run Generator a Level of Effort (LOE)** activity in P6.
- ❑ That seems to sit better because running a generator isn't a deliverable;
it's an **ongoing thing** and exemplifies the purpose of an LOE activity.



Logic – Predecessor and successor logic: Start to finish

- ❑ **Most examples** of start to finish relationships always seem to refer to non-deliverable type activities. The one I see most is **“shift B cannot finish until shift A starts”**.
- ❖ Shift changes and even running generators are ongoing, **non-deliverable items**.
 - ❖ They **don't really represent activity** in the true sense of building a project deliverable.
 - ❖ There is **almost always an easier and more understandable way to model** what SF is supposed to be modeling, particularly when using a tool as flexible as Primavera P6.

Logic – Predecessor and successor logic: Recommended approach

PMI Best practice

Recommended Practice

It is recommended that, once the scope has been drilled down to the lowest level of detail (the activity level), the sequence of work performance is then defined, resulting in relational logic which contains no open-ended activities (other than the start and completion of the project), and where the majority of the activities within the project network are logically tied with finish to start relationships when at all possible.

Prior to sequencing a list of activities, consensus should be reached with the project team, or the “experts” who will be responsible for implementing and managing the successful completion of the project.

“Expert knowledge” is very valuable in developing a schedule/tool that the project team will rely on and use.

Logic – Predecessor and successor logic: Recommended approach

AACE Document

AACE recommended practice for developing logic is:

Development of activity logic is a pure planning step where relationships are established regardless of the dates on which activities fall. The next step in the planning and scheduling process is to consider other factors such as time, resources, and milestones to achieve project objectives. (AACE, 2012, p. 2)

Logic – Predecessor and successor logic: Recommended approach

GAO

- ❑ The **majority of relationships** within a **detailed schedule** should be **finish-to-start**.
- ❑ **Finish-to-start** relationships are **intuitive** because **most work** is accomplished **serially** in that order.
- ❑ **F–S relationships** are implemented **most easily where** work is **broken down** to **small elements**.

Logic – Predecessor and successor logic: Recommended approach

- ❑ Start-to-start and finish-to-finish relationships, in contrast, imply **parallel or concurrent work**.
- ❑ S–S and F–F are **valid techniques** for modeling the **overlapping of activities**.
 - ❖ They may be more **predominant** in schedules whose **detail has not yet evolved**.
 - ❖ However, an **overabundance** of these relationships in **detail schedules** may suggest an overly optimistic or unrealistic schedule.
 - ✓ Particularly in a **detailed schedule**, their **overuse** may complicate the **identification** of the **critical path**.

*S–S and F–F relationships are also prone to **producing unintentional** “dangling” relationship logic, an error that we **describe later**.*

Logic – Predecessor and successor logic: Recommended approach

PMI Best practice

Default Condition

The default relationship type should be FS, with a minimal usage of SS and FF relationship types, and a rare usage of the SF relationship type.

Best Practices

The preferred application of project logic development is the usage of FS relationships between activities, when at all possible, and the adherence to only allowing two open-ended activities: on the start and completion activities/milestones.

Logic – Predecessor and successor logic: Recommended approach

DCMA

The metric 'Logic' provides the possibility for measurement of the '%' of incomplete tasks with no predecessors and/or successors, also called 'dangling' activities. For this metric, it is preferable that the number of unfinished activities missing predecessors and/or successors don't exceed 5%.

The best practice for avoiding incompleteness of tasks is ensuring that every task has at least one finish-to-start or start-to-start relation as its predecessor and at least one finish-to-start or finish-to-finish as its successor. This metric is important in investigating how project tasks are connected to each other and in assuring that no incomplete tasks affect the project schedule negatively.

4. **FS Relationships** – Even though Primavera P6 and Deltek Acumen Fuse both support all relationship types, the DCMA assessment states 90% (or more) of schedule dependencies should be Finish-Start (FS). Start-Start (SS) are acceptable, but building a whole schedule using SS is obviously unacceptable.

$$\% \text{ FS Relationships} = \frac{\# \text{ of logic links with FS Relationships}}{\# \text{ of logic links}} \times 100$$

Logic – Incomplete and dangling logic

- ❑ All activities in a schedule network should have at least **one predecessor** and **one successor**
 - ❖ *There are two exceptions: start and finish milestones*
- ❑ Activities without a predecessor or a successor are called **open-ended activities**
 - ❖ Open-ends can create **inaccurate** amounts of **float** and **errors** in the **critical path**
- ❑ Even if an activity has **predecessor** and **successor** logic relationships, **incorrect or incomplete logic** can arise.
- ❑ Networks should be assessed for **circular logic**—that is, logic that forces **two activities** to be **dependent** on **each other**.

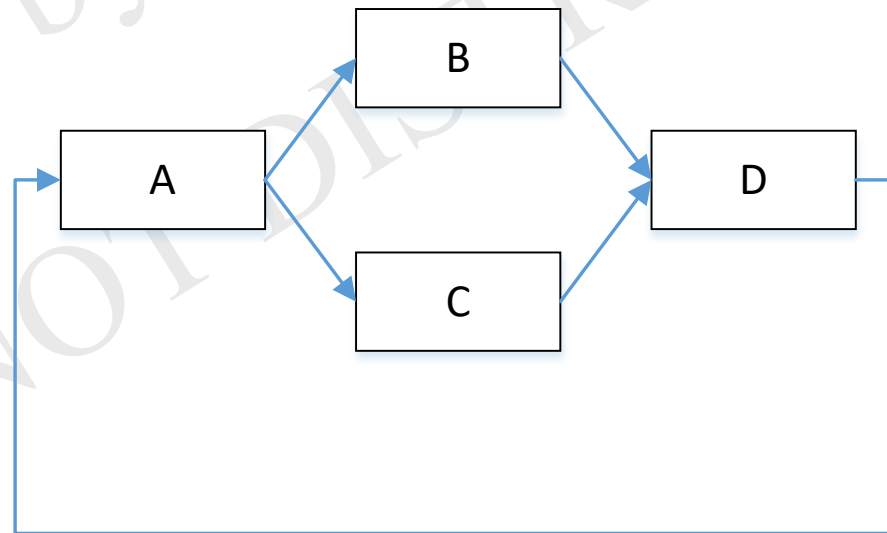
Logic – Predecessor and successor logic: Summary logic

- ❑ Certain scheduling software packages include summary activities as an **option**.
 - ❖ If used, summary activities should not have logic relationships because their **start and finish dates** are derived from **lower-level activities**.

No need for logic relationships on a summary activity in a properly networked schedule

Logic – Incomplete and dangling logic: Loops

- ❑ **Circular logic** creates an **endless loop of work**.
- ❑ When **two or more activities** are **linked** in a **circular manner**, the **logic cannot** proceed from the **beginning** of the network to the **end** of the network.
- ❑ **Loops** are **errors** in logic and must be **fixed** before the network can be calculated properly.



Logic – Incomplete and dangling logic: Dangling or hanging logic

- ❑ Dangling logic is **scheduling logic** with an **improper tie** to an activity's start or end date.
- ❑ Dangling logic is far **more dangerous** for detail activities than milestones.
- ❑ Dangling logic, a form of **incomplete logic**, can **interfere** with the valid forecasting of scheduled activities.
- ❑ Activities with S-S or F-F relationships should be checked for **two types of “dangling” or “hanging” logic**:
 1. Start-date dangling logic
 2. Finish-date dangling logic

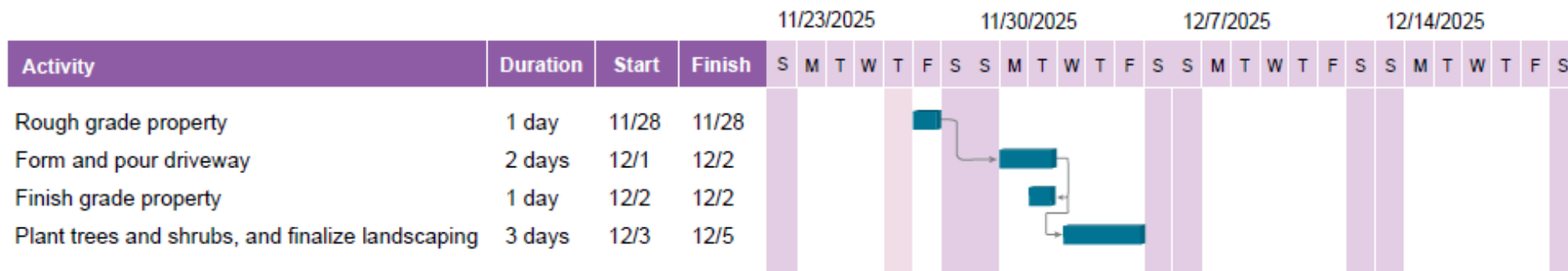
Logic – Incomplete and dangling logic: Dangling or hanging logic

1. Start-date dangling logic:

- ❖ Occurs when an **activity** has a **predecessor and a successor** but its start date **is not properly tied** to logic.
- ❖ In other words, **no preceding activity** within the schedule is **determining the start date of the activity**, with **either** its **start (S-S)** or **finish (F-S)**.

Logic – Incomplete and dangling logic: Dangling or hanging logic – Example

Start-Date Dangling Logic



Source: GAO. | GAO-16-89G

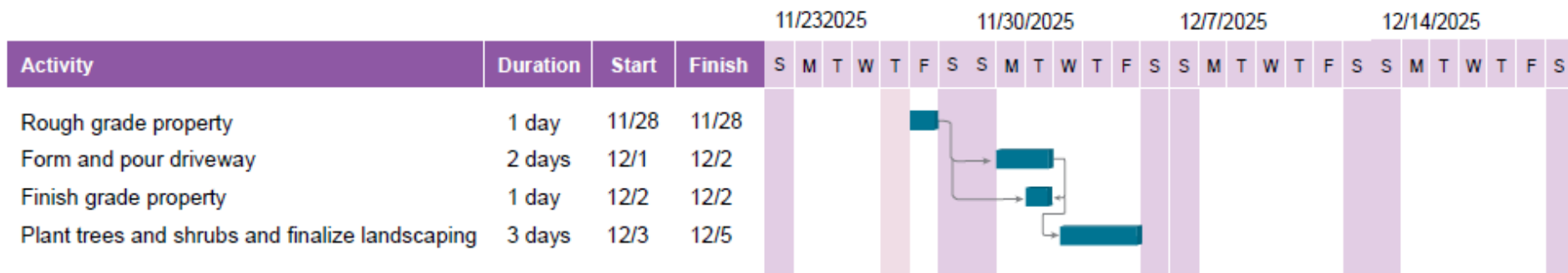
❑ The problem here is that the start date of **“finish grade property”** is not determined by any relationship.

❖ If **“form and pour driveway”** finishes but **“finish grade property”** runs longer than planned, the only resolution—according to the original plan—is for **“finish grade property”** to start earlier.

This is a logical solution but practically impossible because we don't know whether it takes longer or not and how long further?

Logic – Incomplete and dangling logic: Dangling or hanging logic – Example

Start-Date Dangling Logic Corrected with Predecessor Logic



Source: GAO. | GAO-16-89G

- ❑ “**Finish grade property**” should have at least one F–S or S–S predecessor link alongside the F–F predecessor relationship to determine its start date.
- ❑ This requirement might make the **scheduler** to:
 - (1) Find different predecessor or
 - (2) Break down “form and pour driveway” into **two activities**.

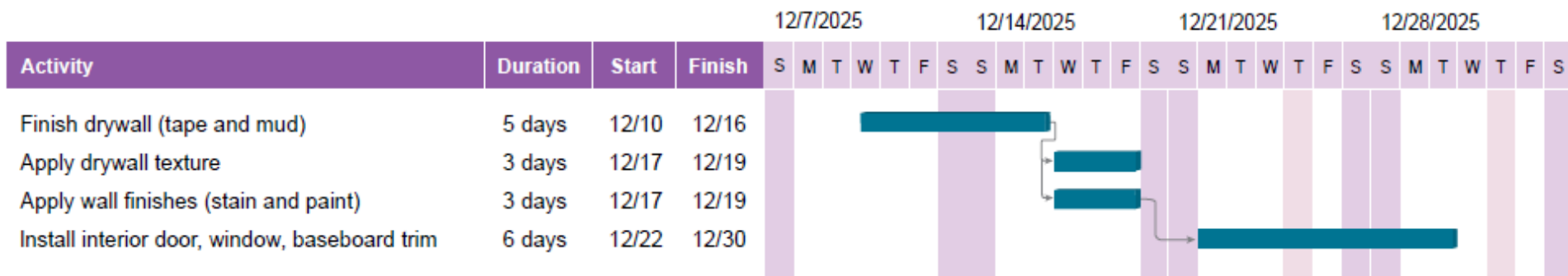
Logic – Incomplete and dangling logic: Dangling or hanging logic

2. Finish-date dangling logic:

- ❖ Occurs when an **activity** has a **predecessor and a successor** but its **finish date is not properly tied** to logic.
- ❖ In other words, **no succeeding activity** within the schedule is **determining the finish date of the activity**, with **either** its **start (F–F)** or **finish (F–S)**.

Logic – Incomplete and dangling logic: Dangling or hanging logic – Example

Finish-Date Dangling Logic



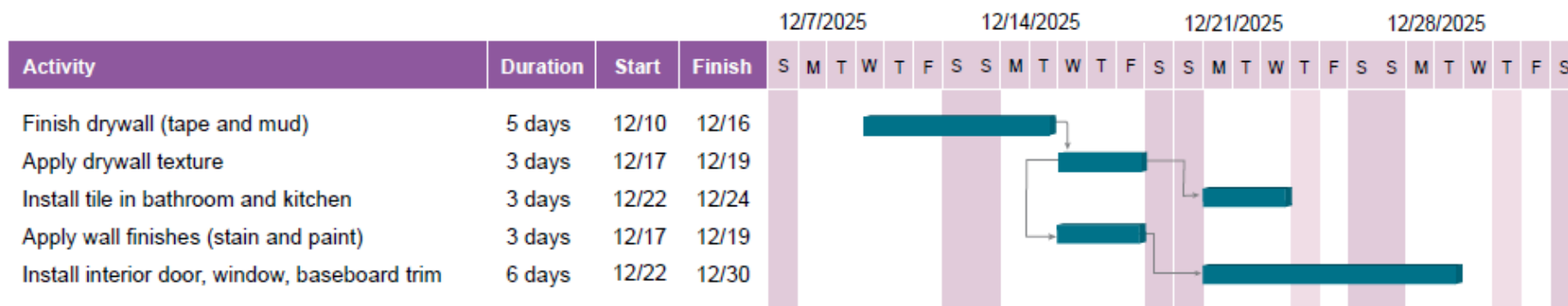
Source: GAO. | GAO-16-89G

- ❑ “Apply drywall texture” has an F–S predecessor link to “finish drywall” and an S–S successor relationship to “apply wall finishes.”
- ❑ While “apply drywall texture” has a successor, its finish date is not related to any subsequent activity.

In other words, “apply drywall texture” can **continue indefinitely** with **no adverse effect** on subsequent activities.

Logic – Incomplete and dangling logic: Dangling or hanging logic – Example

Finish-Date Dangling Logic Corrected with Successor Logic



Source: GAO. | GAO-16-89G

- ❑ An F–S predecessor has been added from “**apply drywall texture**” to “**install tile in bathroom and kitchen.**”

Now, according to the logic model, “**apply drywall texture**” cannot be delayed **without also delaying** “**install tile in bathroom and kitchen.**”

Logic – Important note

PMI Best Practice

Advisories

- Minimize use of finish-to-finish and start-to-start relationship types.
- Rarely, if at all, utilize the start-to-finish relationship type.
- Be careful with the use of SS relationships because the predecessor is in control of all the SS successors.
If the predecessor SS activity is actualized out of sequence, it will allow all other SS successors to start immediately, whether or not it is reasonable. Be careful to review the predecessors to any activity that will have SS successors to ensure that the relationships should not more appropriately be FS from the predecessor.

Which one is **better** FS with negative lag or SS with positive lag?

I will tell you later!

Logic – Important note

DCMA

Logic

The metric 'Logic' provides the possibility for measurement of the '%' of incomplete tasks with no predecessors and/or successors, also called 'dangling' activities. For this metric, it is preferable that the number of unfinished activities missing predecessors and/or successors don't exceed 5%.

$$\% \text{ Missing Logic} = \frac{\# \text{ of tasks missing logic}}{\# \text{ of incomplete tasks}} \times 100$$

Logic – Important note

PMI Best Practice

Purpose

Driving relationships control the start of the successor activities and are relevant in determining the critical path.

Default Condition

All activities, except the first, must have at least one driving relationship.

Advisories

Relationships that utilize SS or FF relationships or FS ties with negative lags may not result in effective driving relationships. (See Section 3.7.3, Use of Lags, for further information.)

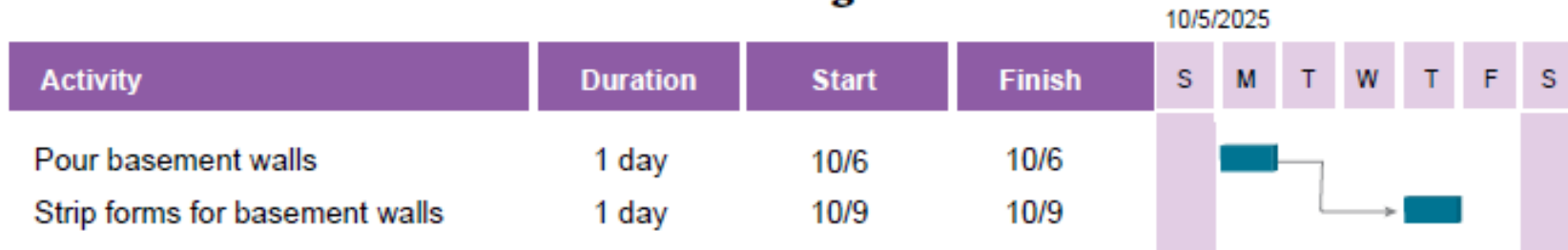
Let's see!

Lags – Definition

❑ **Delay** of the **successor activity** and is between the **finish** of the **first activity** to

1. **Start** of the **second activity**
2. **Finish** of the **second activity**

A Lag



Source: GAO. | GAO-16-89G

Lags – Definition

PMI Best Practice and AACE

Guidelines

The term “lag” is most often used in construction scheduling but the term “lead” is also used. They are interchangeable. Below is a list of definitions from various construction documents and sources for both terms.

Definitions

Lag—Time that an activity follows, or is delayed from the start or finish of its predecessor(s); sometimes called an offset. A lag may have a negative value tied to the finish of a previous activity, reflecting a fast track approach. However, the use of negative lags when building baseline schedule models is poor technique and often prohibited by specification (AACE, 2004, p. 59).

Purpose

Lags can represent a naturally occurring span of time that does not need to be monitored, is not generally a work activity, and is not generally tracked.

Lags can be used for non-work time or mobilization time between activities.

Lags – Definition: Example

PMI Best Practice

Laying pipeline—Excavate, then begin laying pipe a day or two later. The time between excavating and laying pipe could be to mobilize the pipe-laying crew, or to allow the ditches to dry sufficiently for pipe.

Concrete—Excavate for footer, but prior to the completion of footing excavation, start formwork. The lag would represent the amount of time that the excavation crew needed to advance ahead of the formwork crew, to allow the formwork crew to start work and progress efficiently.

Do not use a lag for something like cure time; use an activity instead.

This is what SCL says!

constraint is influential. Lags may be introduced for non-work periods (such as curing of concrete) but better visibility and understanding is provided if such matters are shown as activities in themselves (See Appendix A for definitions of logic links and lags). Activities to be executed by the use of overtime and/or

Lags – Don'ts

GAO

- ☐ Lags must represent a **real need** to delay time between **two activities**.
- ☐ Lags must be **justified** by **compelling reasons** in the **schedule documentation**.
- ☐ When **schedule** contains **many lags**, and **changes** occur to the schedule, the schedule should be **updated manually** and it **defeats** the purpose of a **dynamic schedule** and can cause an **error**.
- ☐ In particular, **F–S relationships** with lags are generally **not necessary**.
 - ❖ If **used improperly**, lags can **distort float calculations** in a schedule and can **corrupt** the calculation of the **critical path**.

Lags – Don'ts

❑ Lags should **not represent**:

1. Procurement activities
2. Other types of work performed by **parties external** to the schedule.

WHY?

❖ Because lags are **not easily monitored**, and do **not respond** to the **availability of resources**. WHY?

✓ Because lags are **static** and **cannot** be **updated** with **progress**.

Do not use lags as buffers or margin between two activities for risk.

Lags – Don'ts

- ❑ **Lags** are **often misused** to **force successor** activities to **begin** on **specific dates**.

The critical path should be **free** of lags because they make the identification and management of critical activities **ambiguous**.

WHY?

1. **Evaluating** a **critical path** that includes **lags** can be **difficult** because the **lag** may or may not represent **work** that has to be **accomplished** by some resource.
2. **Because** lags are **not represented** in a schedule as an activity, they can **easily be overlooked** as drivers of the **finish milestone date**.

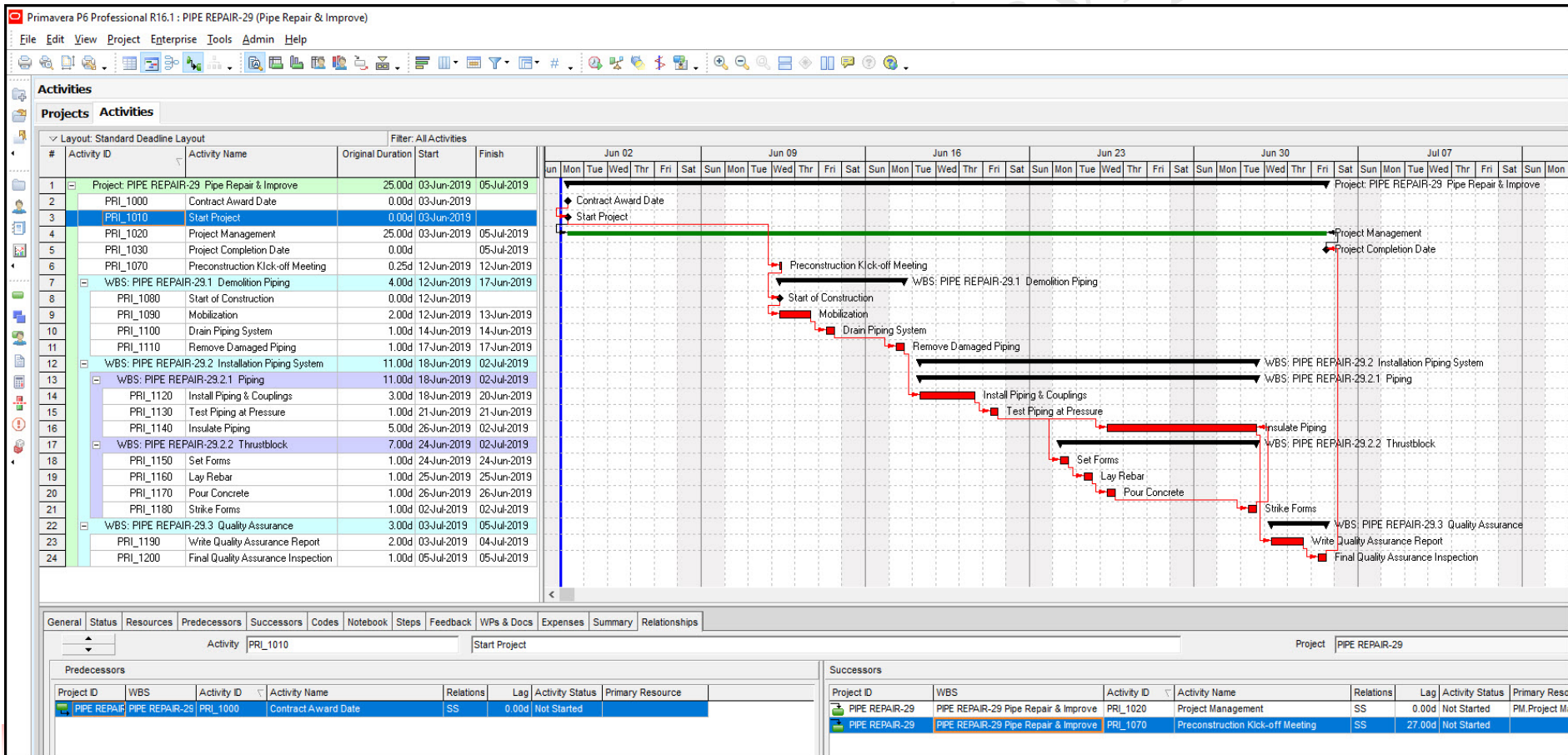
Let's take a look!



Lags – Don'ts

فرآیند یکپارچه برنامه ریزی، زمانبندی، ارزیابی و کنترل پروژه
استاندارد اجرایی: اولویت بندی و تعیین روابط بین همه فعالیتها

- In this schedule we want the **pre-construction kick-off meeting** to take place on 12-June-2019. To achieve this date we **insert a lag** between start project and preconstruction kick-off meeting.



Why this is problematic?

Lags – Don'ts

Why this is problematic?

What takes place during this waiting period?

Our schedule is not very transparent, which affects its overall quality.

*Worse still if the start project milestone **delays for any reason** the lag between start project and preconstruction kick-off meeting will **not adjust**, accordingly.*

This is a problem.

*Any updates in the predecessor activity will make **preconstruction kick-off** meeting miss our planned 12-June-2019 date.*

Lags – Don'ts

DCMA

Lags

The metric 'Lags' is the opposite of 'Leads'. Specifically, it allows an evaluation of the '%' of project activities that have a positive lag between each other. The metric actually indicates the time (hours/days) calculated when a certain activity starts after the finish date of its predecessor.

The DCMA threshold for this metric is more flexible than the one for the Leads and it advises that the lags in the project schedule shouldn't be greater than 5%. However, the excessive use of lags as well can cause an obstruction to the critical path analysis, therefore it is recommended that they are avoided. A good practice for this metric is to present activities with specific names rather than with an 'X-days lag'.

$$\begin{aligned} & \% \text{ Positive Lags} \\ &= \frac{\# \text{ of logic links with positive lags}}{\# \text{ of logic links}} \times 100 \end{aligned}$$

Lags – Don'ts

- ☐ Avoid using lags in place of tasking representing **budgeted work**.
- ☐ Avoid using lags in place of tasking representing **resources**.
- ☐ Avoid using lags in place of tasking representing **status of the work**.
- ☐ Avoid using lags in place of tasking representing **contingency**.
- ☐ Avoid using lags in place of tasking representing **procurement, review, submittal, etc.**
- ☐ Avoid using lags in place of tasking representing **external work**.
- ☐ Avoid using lags in place of tasking representing **constraint**.
- ☐ Avoid using lags in place of tasking representing **needs to be monitored**.

Schedule that has many lags indicates higher ambiguity

Lags - Dos

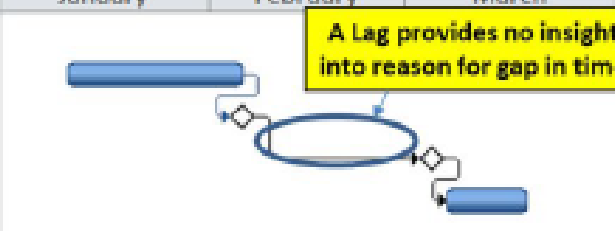
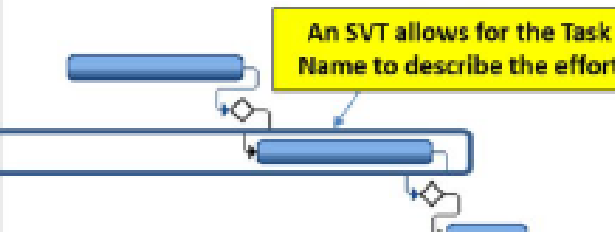
- ❑ A **lag** that indicates a **waiting period**, such as for document review or materials delivery,

should be **replaced** with an activity that can be

1. Actively monitored
2. Statused
3. Perhaps mitigated if necessary

Also,

*Consider whether it is more appropriate to use a **Soft Constraint** versus a **Lag***

	Task Name	Duration	Predecessors	1st Quarter		
				January	February	March
1	Lag (Govt. Review Time is less visible)	50 days				
2	Draft NAVSYS TRD	20 days				
3	Deliver Draft NAVSYS TRD to Government	0 days	2			
4	GFE - Receive Govt. Comments to Draft NAVSYS TRD	0 days	3FS+20 days			
5	Incorporate Govt. Comments to Draft NAVSYS TRD	10 days	4			
6						
7	SVT (Govt. Review Time is more visible)	50 days				
8	Draft NAVSYS TRD	20 days				
9	Deliver Draft NAVSYS TRD to Government	0 days	8			
10	SVT - Government Review of Draft NAVSYS TRD	20 days	9			
11	GFE - Receive Govt. Comments to Draft NAVSYS TRD	0 days	10			
12	Incorporate Govt. Comments to Draft NAVSYS TRD	10 days	11			

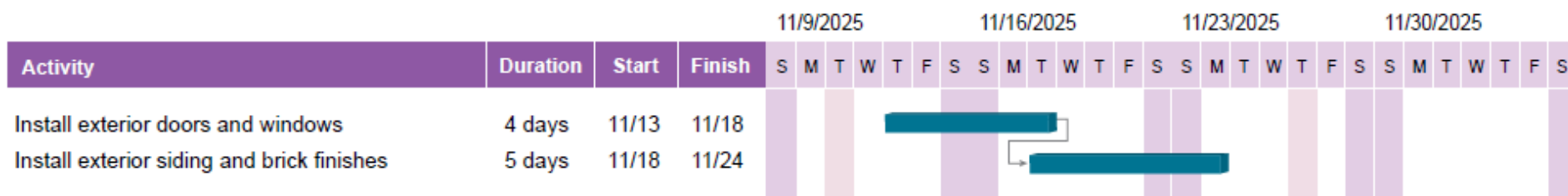


Leads – Definition

❑ **Acceleration** of the **successor activity** and can be used only on **finish-to-start** **activity relationships**.

❖ **Negative lag**, known as a lead, is used to **accelerate a successor activity**.

A Negative Lag (or Lead)

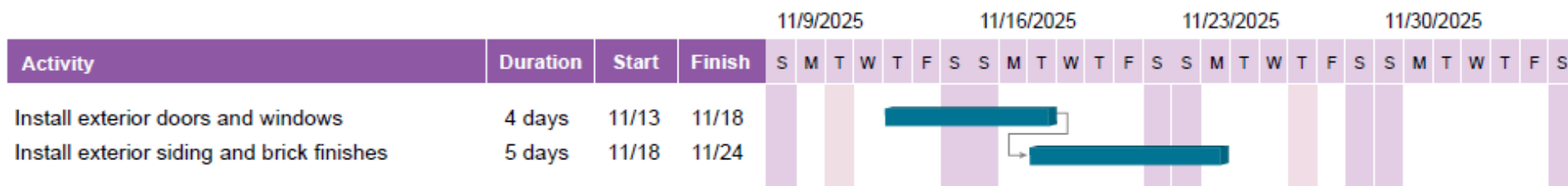


Source: GAO. | GAO-16-89G

Leads – Don'ts

- ❑ Leads, in particular, are **often unnecessary**.
- ❑ As **negative lags**, leads imply the **unusual measurement** of **negative time** and **exact foresight** about **future events**.
- ❖ Delay in **installing doors** within the last day may **cause immediate problems**.

A Negative Lag (or Lead)



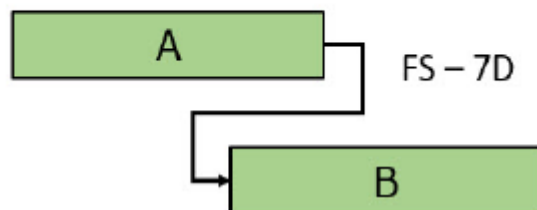
Source: GAO. | GAO-16-89G

*But leads of **weeks** or **even months** are unfortunately common in some schedules, questioning people's credibility when forced to see **that far** into the **future on a risky project**.*

Do you want to see an example?
Let's see!

Leads – Don'ts

Finish to Start w/ Lead



Task B can start 7 days before Task A finishes

Figure 3

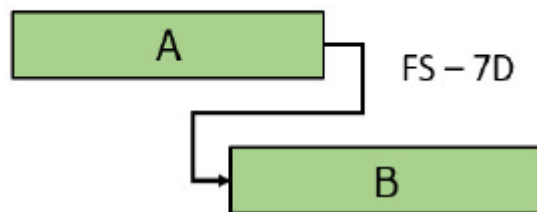
tensix.com

Schedulers like lead as it is a way to fast track the schedule by performing a portion of each activity in parallel. And sometimes a lead modifier seems to better fit the activity narrative. For instance the schedule that says 5-days before you receive equipment on site you begin mobilization. That appears logical. You know the delivery date, so 5-days before that date you mobilize your material, and then crew. It's a nice fit between mobilize and receive equipment on site.

**So far so good. Right?
Let's see!**

Leads – Don'ts

Finish to Start w/ Lead



Task B can start 7 days before Task A finishes

Figure 3

tensix.com

But what happens if your delivery date is delayed? Oops! In that case you have your crew ready to proceed, but they are left waiting in limbo for the equipment to arrive on site. Not good! Thus the true story of negative lag is revealed. Negative lags imply the unusual measurement of negative time and require an exact foresight about future events.

Again, for the **Finish-to-Start relationship with negative lag** you are saying **commence** **successor activity** a period of time **before** the **predecessor** is **100% complete**.

Leads – Don'ts

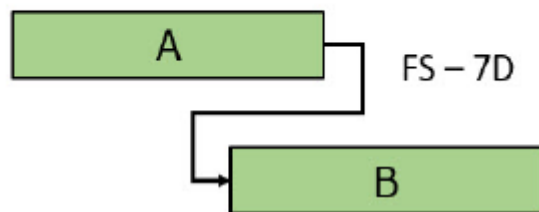
❑ Link the start of another activity based on this end prediction is not good scheduling practice.

❖ So you are deciding whether to **start an activity now** based upon a future activity situation, which is not ideal.

How about **total float**?

Leads – Don'ts

Finish to Start w/ Lead



Task B can start 7 days before Task A finishes

Besides the required future foresight fundamental flaw with negative lag the DCMA 14-point assessment says that leads may distort total float and adversely affect the critical path. The DCMA assessment does not provide any examples of float distortion.

Figure 3

tensix.com

This is exactly the reason that
although DCMA **allows** for positive lags (5%),
it does **not tolerate** negative lags!

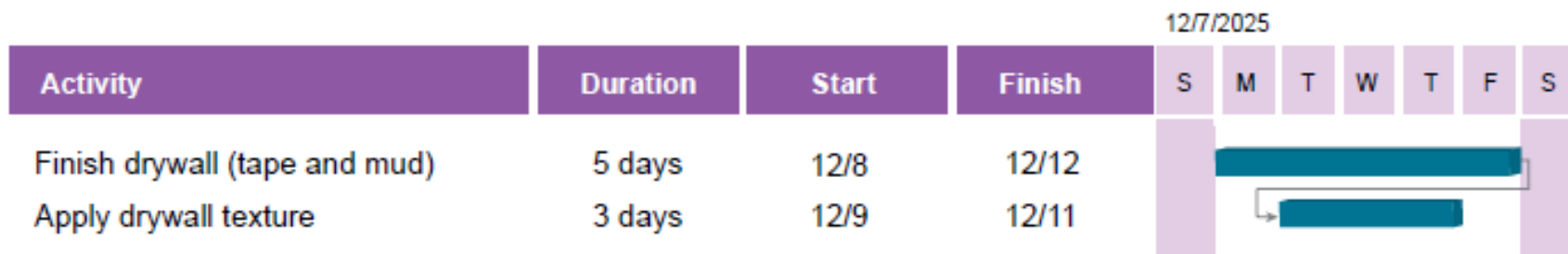
SO WHY THIS IS IMPORTANT?
YOU WILL SEE LATER IN THE **SOLUTION**!

First let's see **why total float**?

Leads – Don'ts – Example

- ❑ Using **leads improperly** can cause logic failures when a **lead** is **longer than the successor activity**.
- ❑ Applying texture takes only 3 days, which means this activity will actually finish before the drywall is finished.
 - ❖ This causes a **conflict** between the **lead** and the **finish-to-start relationship**.
 - ❖ Because the **lead** causes texture application to **finish a day earlier** than the finishing of drywall, an **artificial day of float** may be introduced into the **sequence of activities**.

Logic Failure Associated with a Lead



Leads – Don'ts – Example

Let's see this analysis in real example!

Leads – Don'ts – Example

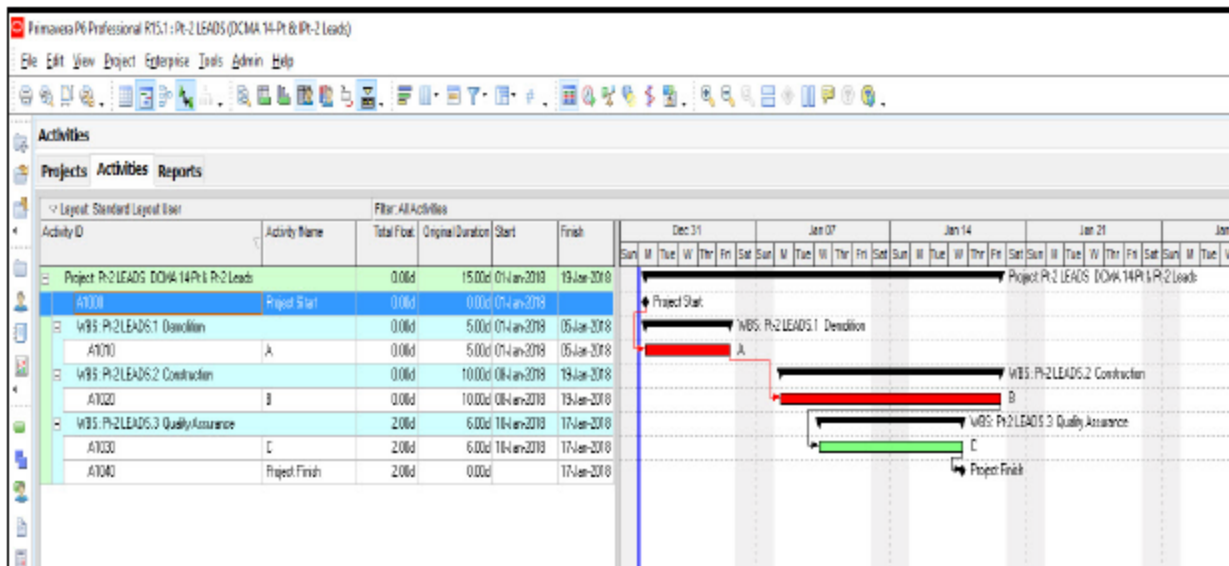


Figure 4

In Figure 4, the lead spread between B and C activities is so wide that the successor activity C actually finishes before the predecessor activity B. This is a clear violation of network logic. Why define a FS relationship between activities B and C, if you let the schedule dishonor it?

Leads – Don'ts – Example

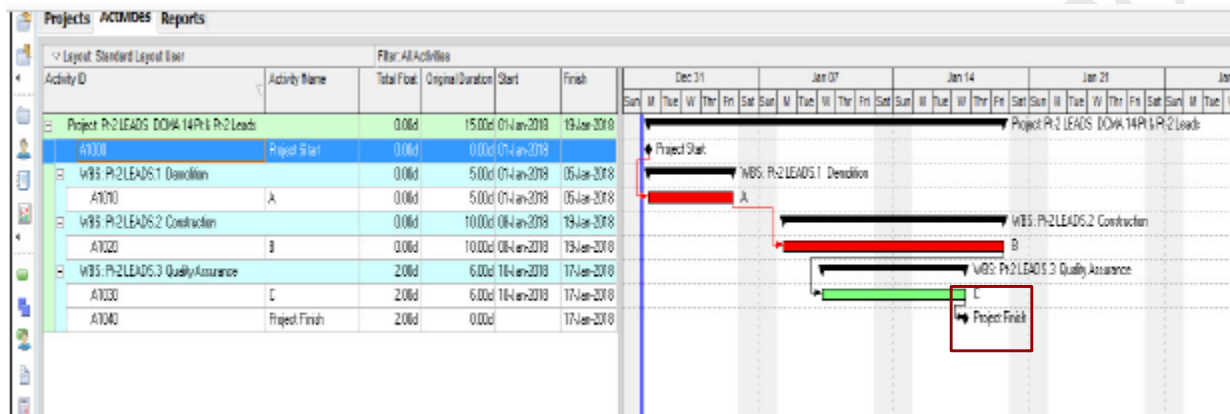


Figure 4

Note that activity C and activity 'project finish' both have a total float of 2-days, which means they can both be delayed 2-days without delaying the entire project. Well, 'project finish' is the end of the project, so these two activities, C and milestone 'project finish' should have 0-days total float.

Leads – Don'ts – Example

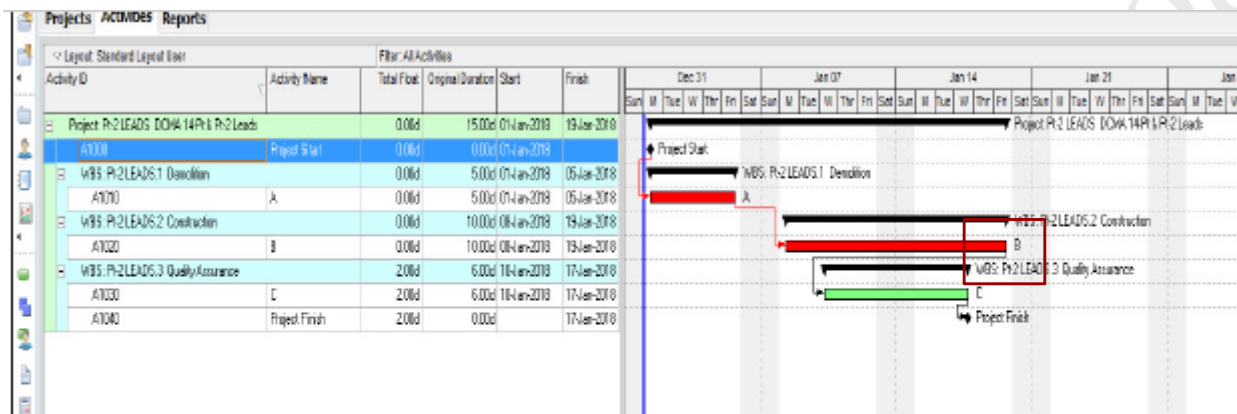


Figure 4

Also note, activity B extends beyond the project finish date, but has 0-days total float when logic would say it should have minus 2-days total float. Note further that activity C is not on the critical path, so we are additionally having issues with our critical path.

Leads – Don'ts

PMI Best Practice

Negative lags can hamper risk analysis in some software. Negative lags may prevent the project team from being aware of the underlying "logic" being used to represent the activities, as it is counterintuitive. Negative lags, when used in preliminary schedules, carry a high risk of forcing the successor to start earlier than the predecessor during schedule duration adjustments.

How to solve?

Leads – Dos

HOW TO SOLVE?

GAO says:

The concept of a **lead is better represented** by a **positive lag on an S–S relationship** or, even **more straightforward**, on **F–S relationships with no lags** using activities of shorter duration.

❑ Two solutions:

1. **Use** an **S–S relationship** with **positive lag**

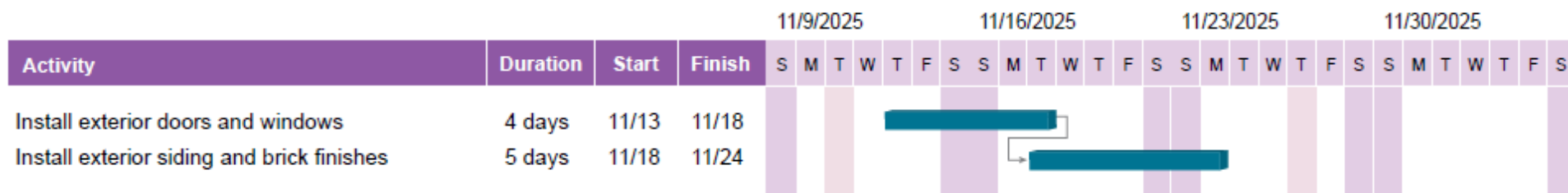
or, even more straightforward,

2. **Breakdown** the **activity** and **use** **F–S relationships**

Leads – Dos: FS Example

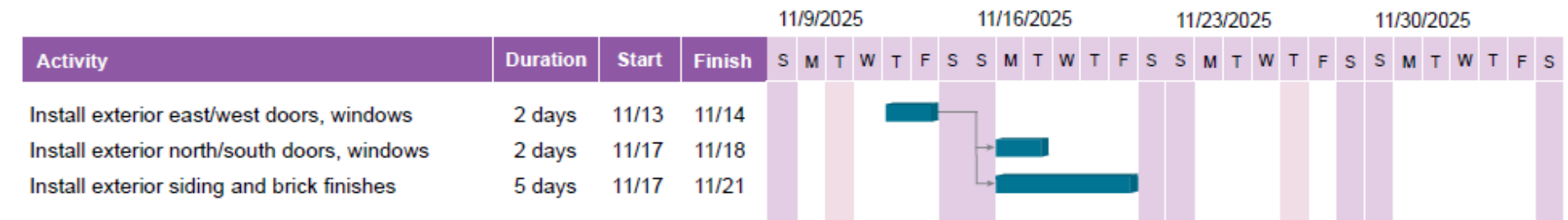
- ❑ For example, instead of a lead, the installing doors and windows activity should be broken into smaller activities to identify the proper F–S activities.

A Negative Lag (or Lead)



Source: GAO. | GAO-16-89G

Eliminating Leads with Finish-to-Start Links



Source: GAO. | GAO-16-89G

Leads – Dos: SS Example

Oracle

❑ Oracle help center says:

- ❖ https://docs.oracle.com/cd/E17266_01/p6help/help/en/helpmain.htm?toc.htm?46290.htm

Negative Lags - Relationships with a Lag Duration of Less Than 0 Section of the Schedule Check Report Dialog Box

Overview

Use this section to identify relationships with negative lags. Negative lags can affect critical path and distort total float. When activity durations change, activities with negative lag can be scheduled unrealistically. Negative lags can be used to schedule overlapping activities. For example, when digging a trench, workers can start laying pipe in the completed sections of the trench while the rest of the trench is still in progress. To avoid the distortion that negative lag can cause, you might consider using positive lag on a Start to Start relationship rather than a negative lag on a Finish to Start relationship.

Leads – Dos: SS Example

❑ Read these article from the following ling:

❖ <https://tensix.com/2015/05/the-negatives-of-negative-lag/>

If at all possible create your schedule such that successor activities are commenced based upon completed predecessor scope of work, such as in the case of a Finish-to-Start with positive lag or a Start-to-Start with positive lag.

*This will provide your schedule with the **transparency** and efficiency required for successful implementation. And you will **avoid** the inefficiencies of **mobilized crew** having to wait for predecessor activities to **complete sufficient scope** of work for the implementation of their successor activity.*

Leads – Dos: SS Example

PMI Best Practice

- Negative lags are counterintuitive to most schedule users, and should not be used in any circumstances.
- Check all SS and FF relationships with lags for possible sequestering.

Leads – Dos: SS Example

DCMA

The leads in the schedule should equal zero, therefore there shouldn't be tasks with a negative lag between each other. If there is a part of the schedule, which could prove itself as a potential problem, the tasks in question should be decomposed into separate tasks. This metric can impede the critical path analysis, therefore it is preferable that is used within the limits.

$$\% \text{ Leads} = \frac{\# \text{ of logic links with negative lags}}{\# \text{ of logic links}} \times 100$$

SEE?

Although DCMA **allows** for positive lags,
it does **not tolerate** negative lags

Leads – Dos: SS Example

So, why some schedulers saying that **SS with positive lag is not appropriate**?

Leads – Dos: SS Example

PMI Best Practice

It is important to note that **progress** in the predecessor is **not required** for the successor to start.

Positive lags can allow the successor to be scheduled simply by the calendar rather than any amount of progress of the predecessor. Once the predecessor starts, an SS lag will automatically schedule start of the successor once the lag value is reached in calendar time, so the modeling will not provide the logic needed to force some amount of progress on the predecessor prior to starting the successor.

This is exactly opposite to the problem we have in FS with negative lags

HOWEVER
It affects less in **delay analysis!**

If you have progress issues, use FS after **breaking down** the activity!

موسسه مهندسی و مدیریت ساخت علوی پور (ACEMI)

دفتر آموزش

تهران، بلوار فردوس شرق، مجتمع آبگینه، طبقه سوم اداری، واحد C42



+۹۸ ۲۱ ۴۶ ۱۰۰ ۴۴۵



+۹۸ ۲۱ ۴۶ ۱۰۰ ۴۵۰



info@dralavipour.com



www.dralavipour.com



@alavipour_pm



@acemi_social



@alavipour_pm



@IRANACEMI

