

Grail



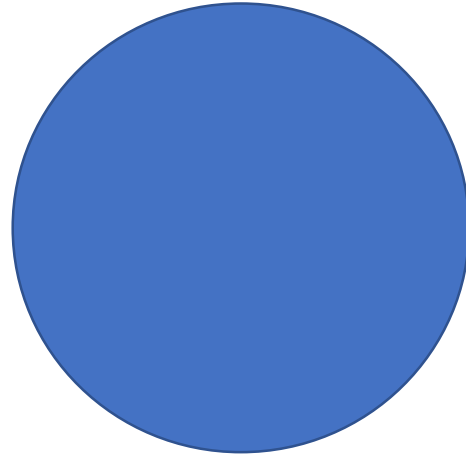
Professional Services

# Time Management

Mark A. Ward

*Lost time is never found again.*

Benjamin Franklin



# An Introduction to Time Management

# Time Management Overview

- Managing time in a project is essentially about creating (or adopting), implementing, and managing a schedule to ensure a project is finished on or before a desired completion date.
- There are several steps associated with this field, and these activities vary by the development process we select to manage a project (e.g, traditional versus agile).

# Activities and Resources

- Time management activities include the creation or adoption and modification of a schedule management plan for a project.
- It also includes the identification of activities and milestones associated with a project.
- In addition, we will identify the key attributes linked with the activities to include the resources that are needed to complete the activities.

# Activities and Sequencing

- Once activities and their resources have been identified, we can sequence them using precedence or conditional diagramming methods.
- We can then calculate the durations of the activities using analogous, parametric, and three-point estimates and the program evaluation and review technique among others.

# Schedule

- From this point, we will create a schedule and determine critical pathways. These pathways are discovered by performing forward and backward passes and float calculations.
- We can also adjust the schedule as needed during its execution by applying leads and lags, conducting what-if analysis (e.g., Monte Carlo simulations), and using other approaches.



# Schedule Modeling

- If a project uses scheduling software to model a schedule, the plan should tell us about the specific version of the software we will use and who will have access to it.

# Schedule Model Parameters

- Some of the more important questions the plan should answer with respect to the scheduling model include:
  - What activity attributes will be modeled?
  - How often will the model be updated?
  - Who will have access to the model?
  - Who has authority to change the model?
  - Who is responsible for updating the model?

# Uniquely Identified Activities

- Activities are uniquely identified in a project using alphanumeric codes or symbols. These uniquely identified activities as well as scheduling data are further linked with cost data to assist us in developing detailed budgets.
- For example, a schedule may identify an activity that uses eight hours of labor as a resource. The hourly labor wage rate for that activity (cost data) can be used to calculate the overall cost of the resource for that activity.

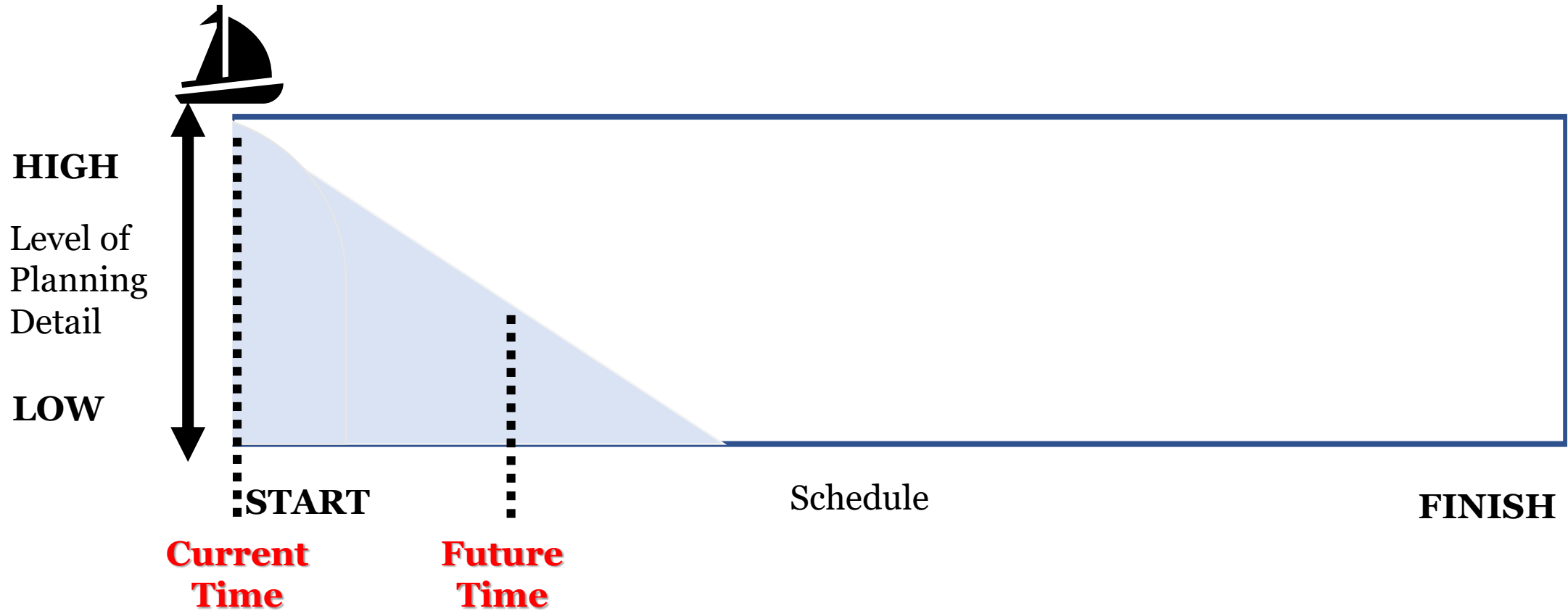
# Schedule Release and Iteration Length

- If a project is being managed using an agile methodology, it is often governed in short (incremental) cycles (iterations).
- End-user value is established (user stories) and is delivered incrementally throughout the life cycle of a project. In this way, the value these deliverables give to the customer is emphasized.
- This approach is a type of rolling wave planning.

# Rolling Wave Planning

- Rolling wave planning is a codification of progressive elaboration. Early phases have relatively detailed planning in a work breakdown structure (WBS). Hence, there are more levels of decomposition and activity identification is more nuanced.
- Later phases are less often less detailed in a WBS, and, therefore, they have fewer levels of decomposition. As a project progresses, detailed planning is accomplished on those parts of a project where the “wave” is approaching (Larman, 2004).

# Rolling Wave Planning Diagram



**Figure 6-1. Rolling Wave Planning Diagram**

# Schedule Demand: Kanban

- Another agile approach is known as the Kanban development method in which work is scheduled in accordance with available resources or capacity (known as pull-based scheduling) as opposed to work being pushed into the schedule (Corey, 2008).
- This approach is thought to improve the management of bottlenecks and is popular in lean manufacturing.

# Schedule Release and Timeboxing

- If a project is being managed using an agile methodology, we may use timeboxes.
- A timeboxed approach\* to a schedule is used when each key deliverable has its own unique in-depth schedule plan (e.g., iteration duration, delivery due date, and cost).

\*Also referred to as a *schedule as independent variable* or SAIV (Boehm, 2004).



# Scope and Timeboxing

- In a traditionally-managed project, scope is fixed at the beginning of a project.
- In agile projects, iteration deadlines are fixed (instead of scope). Hence, project teams prioritize and pick the most important deliverables to work on at the beginning of a project.
- Timeboxing is especially popular in software development.

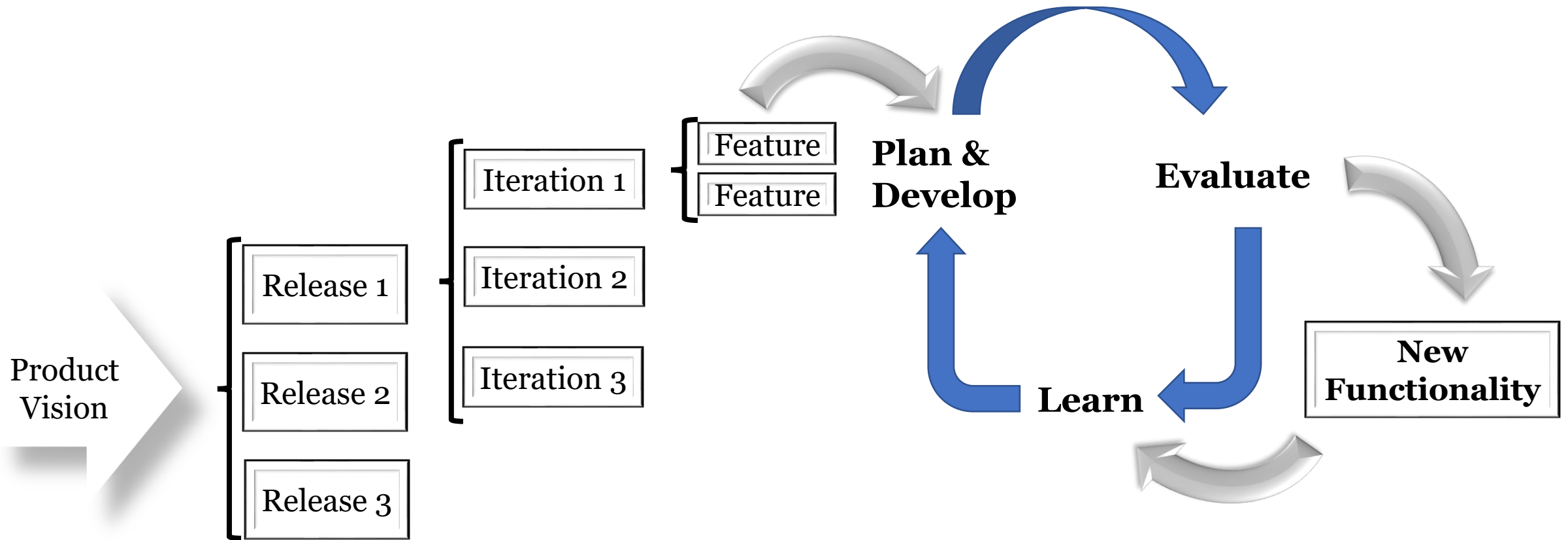
# Agile Release Planning

- Agile release planning provides a high-level summary time line (usually 3 to 6 months) based on a product vision.
- It lists the number of iterations or sprints (usually two weeks in duration).
- It illustrates what features will be available to the customer at the end of each iteration.

# Schedule Adaptive Life Cycle

- Adaptive life cycles are found in agile projects. They reflect the iterative or incremental approach to managing a project.
- We should provide new functionality to the customer at the end of each iteration.

# Schedule Release: Adaptive Life Cycle



**Figure 6-2. Schedule Release: Adaptive Life Cycle**

# Identifying and Listing Activities and Milestones

# Approaches to Identifying Activities

- We need to identify all the work that has to be accomplished to achieve the goals of a project.
- Activities can be identified by enlisting subject matter experts, examining archives, conducting analyses, and using other resources.
- For example, a work breakdown structure with a sufficient level of decomposition can show us what must be created and, therefore, will help us calculate how long it will take to produce a deliverable.

# Activity Duration

- In an effort to make work more manageable, we divide it into smaller parts. In this case, the smaller parts are units of time.
- A rough rule of thumb is to divide the work into activities that are eight to eighty hours long (roughly one to two work weeks). This is the eight/eighty rule.

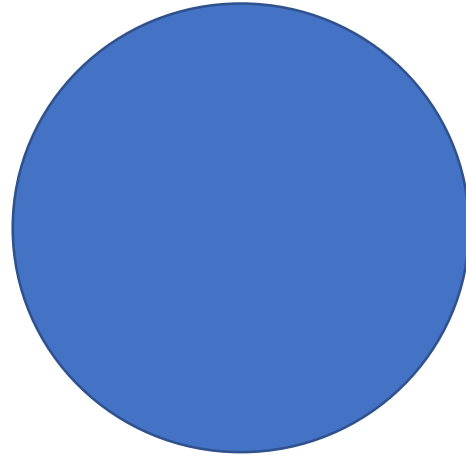
# Activity Characteristics

- After activities have been identified, important characteristics associated with each activity should be ascertained and listed.
- Some of the more common characteristics include activity codes, predecessor and successor activities, logical relationships, leads and lags, resource requirements, imposed dates, and constraints and assumptions.



# Milestone List

- A milestone list categorizes and catalogs all milestones as well as illustrates which milestones are mandatory (i.e., contractual obligation) or optional (i.e., project requisites or historical precedent).
- Note that milestones have no duration. For example, a decision can be made at a certain milestone to continue a project. The decision, itself, has no duration.



# Sequencing Activities

# Sequencing Activities Overview

- Sequencing activities involves placing work in an efficient and effective order with given project constraints (e.g., availability of resources).
- We will employ network diagramming tools to sequence our activities.

# Precedence Diagramming Method (PDM)

- The PDM is a network diagramming system used to show a project's schedule.
- The PDM is also sometimes called a pure logic diagram.

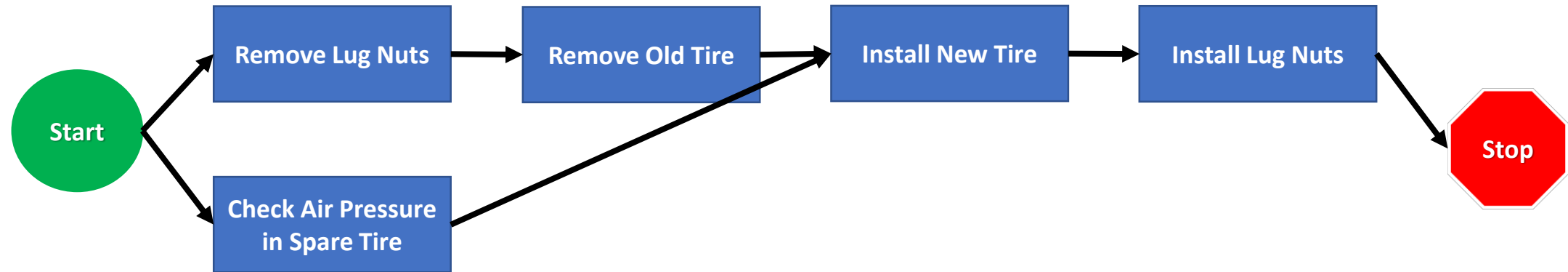
# Nodes

- Activities in the PDM are symbolized by boxes (also called nodes).
- This method is sometimes called activity on node (AON) because activity information is written “on” (inside) the box.
- Logical links join activities by drawing connecting arrows between these nodes.

# Node Information

- Nodes (activity boxes) on a diagram are rendered and read from a reader's left to right.
- Nodes contain at least an activity name and some sort of code that uniquely defines each activity.
- They may also have relevant dates (start, stop), slack time, and other important information.

# Precedence Diagramming Method Example



Note that the length of a connecting arrow does not correspond to the duration of an activity.

Start and Finish nodes are milestones. They signify the beginning and end of the project, but they have no duration.

**Figure 6-3. Precedence Diagramming Method Example**

# Relationships



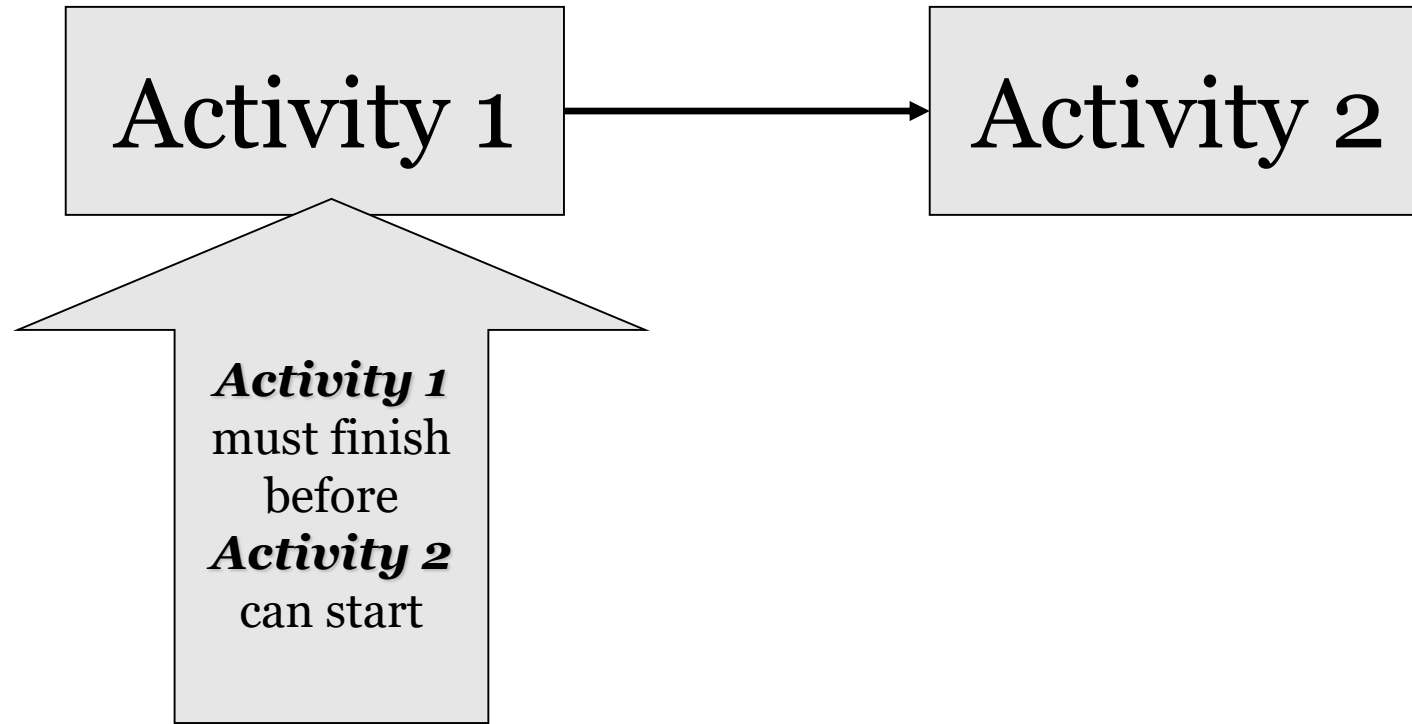
# Relationships Overview

- Relationships reflect sequencing between two or more activities or between activities and milestones.
- These (logical) relationships are often referred to as activity dependencies (not to be confused with project dependencies).
- There are four standard relationships (activity dependencies).
- Multiple relationships can be attributed to a single activity or milestone at the same time.

# Standard Relationship Types

- Finish-to-start (FS)
- Finish-to-finish (FF)
- Start-to-start (SS)
- Start-to-finish (SF)

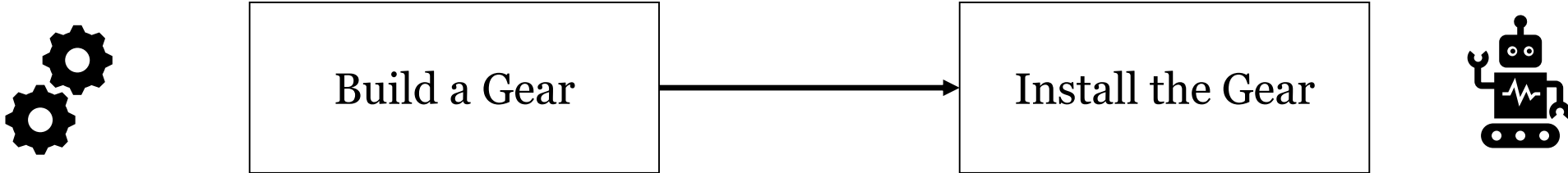
# Finish-to-Start (FS) Relationship



Most often-used activity dependency

**Figure 6-4. Finish-to-Start (FS) Relationship**

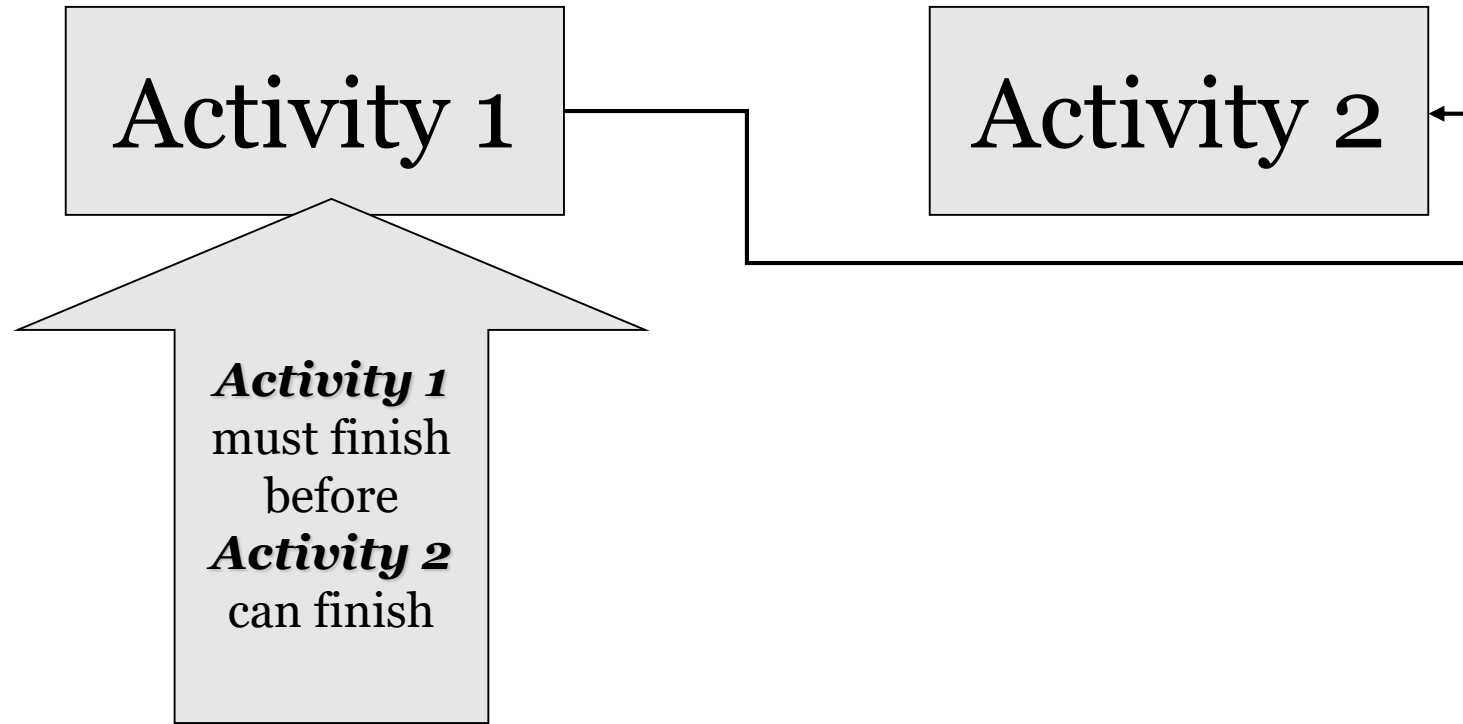
# Finish-to-Start (FS) Relationship Example



- We have to build the gear before we can install it.
- In other words, we have to finish building the gear before we can start installing the gear.

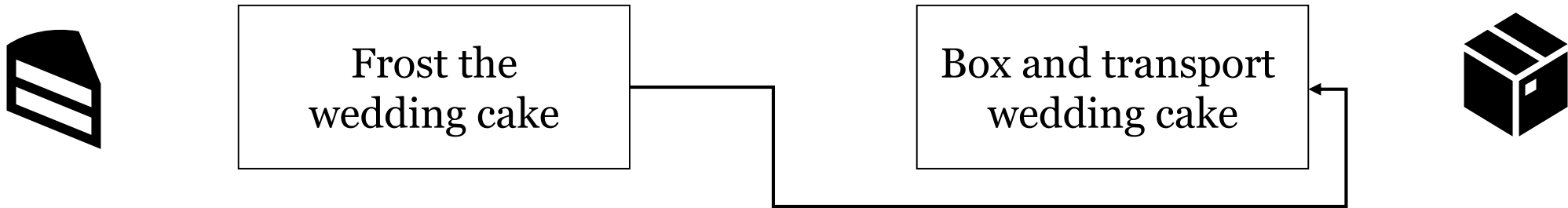
**Figure 6-5. Finish-to-Start (FS) Relationship Example**

# Finish-to-Finish (FF) Relationship



**Figure 6-6. Finish-to-Finish (FF) Relationship**

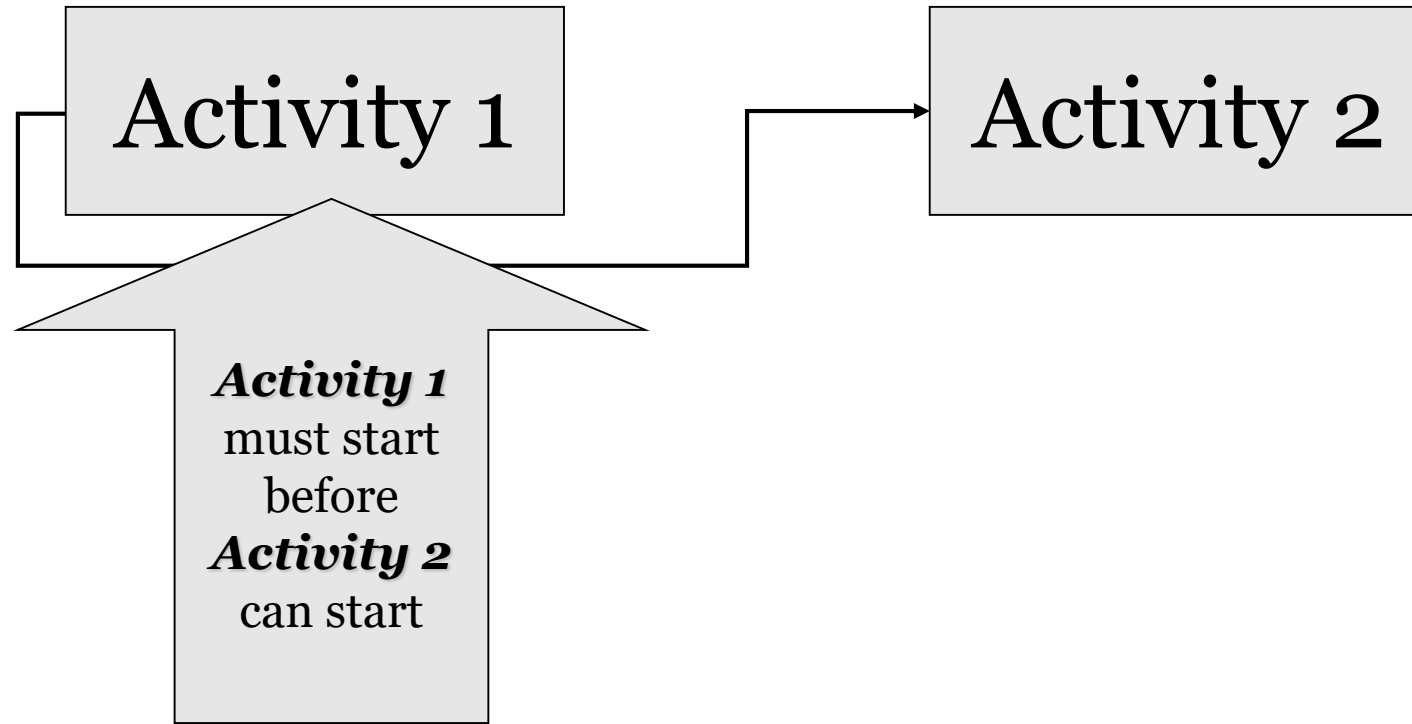
# Finish-to-Finish (FF) Relationship Example



- Before the baker can finish boxing and transporting the wedding cake, the baker must finish frosting the cake.
- This relationship constricts when the successor can finish.

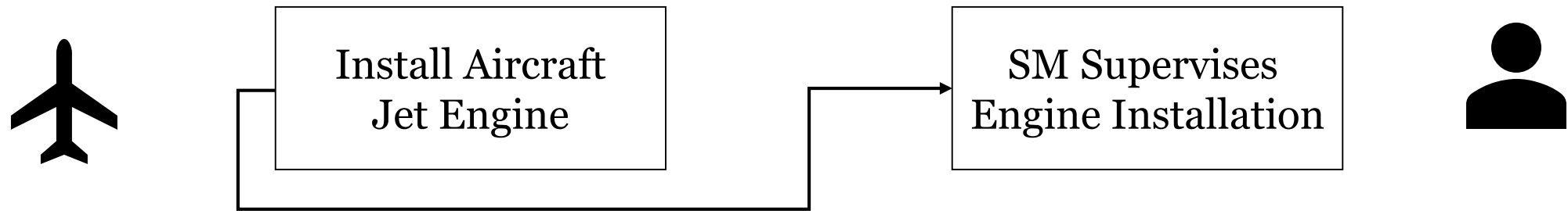
**Figure 6-7. Finish-to-Finish (FF) Relationship Example**

# Start-to-Start (SS) Relationship



**Figure 6-8. Start-to-Start (SS) Relationship**

# Start-to-Start (SS) Relationship Example

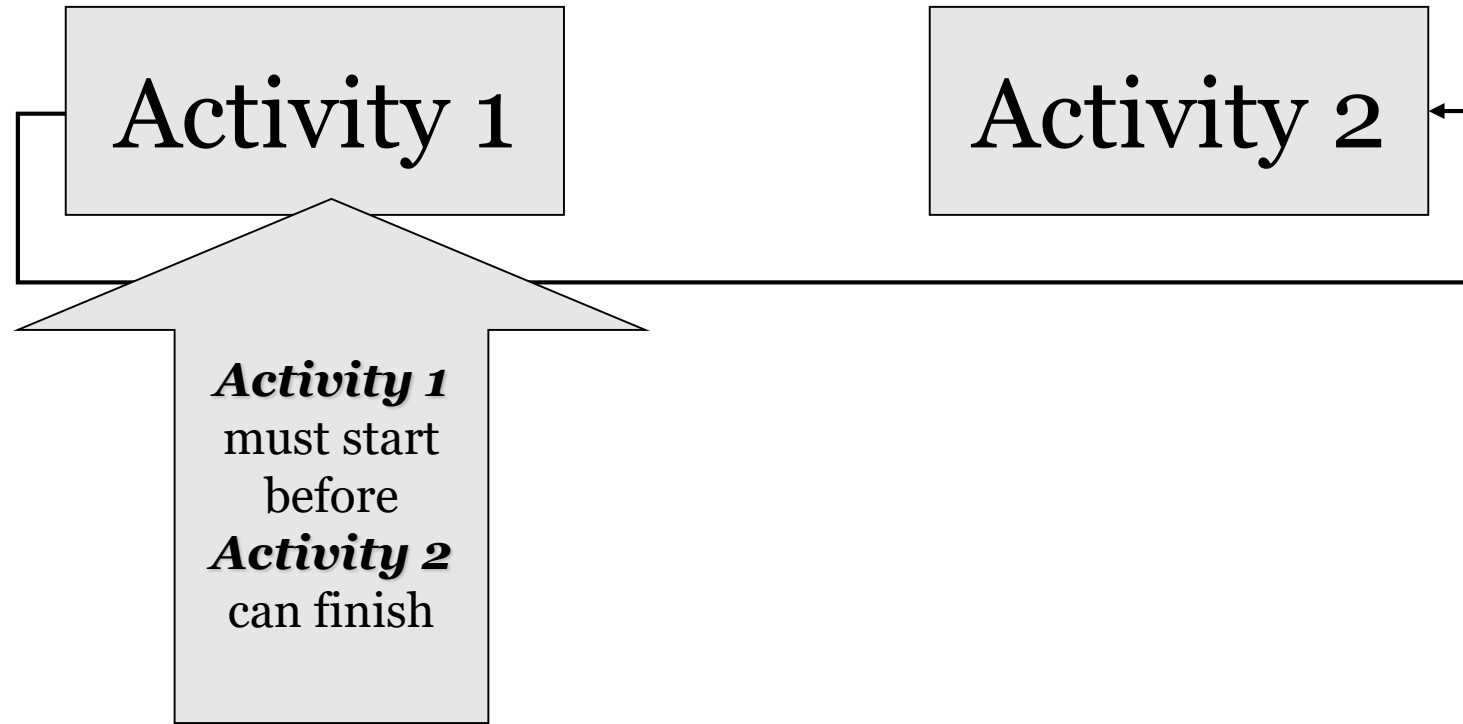


- Before the mechanics begin installing a jet engine, the senior mechanic (SM) must be present to supervise the installation. That is, before the team can start to install the engine, the SM must start supervising the installation.
- This relationship constricts when the predecessor can start.

**Figure 6-9. Start-to-Start (SS) Relationship Example**



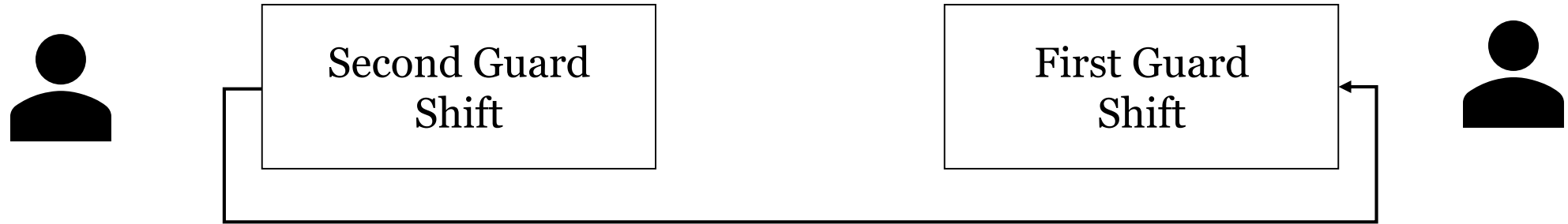
# Start-to-Finish (SF) Relationship



Least often-used activity dependency

**Figure 6-10. Start-to-Finish (SF) Relationship**

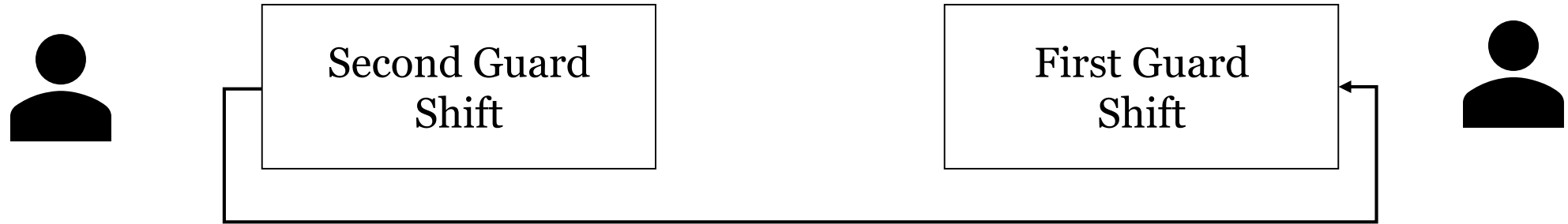
# Start-to-Finish (SF) Relationship Example



- Start-to-finish is used to emphasize that a certain activity (predecessor) must begin before you can finish another activity (successor).
- The relationship states that the second guard shift must start before the first guard shift can finish.

**Figure 6-11. Start-to-Finish (SF) Relationship**

# Start-to-Finish (SF) Relationship Example



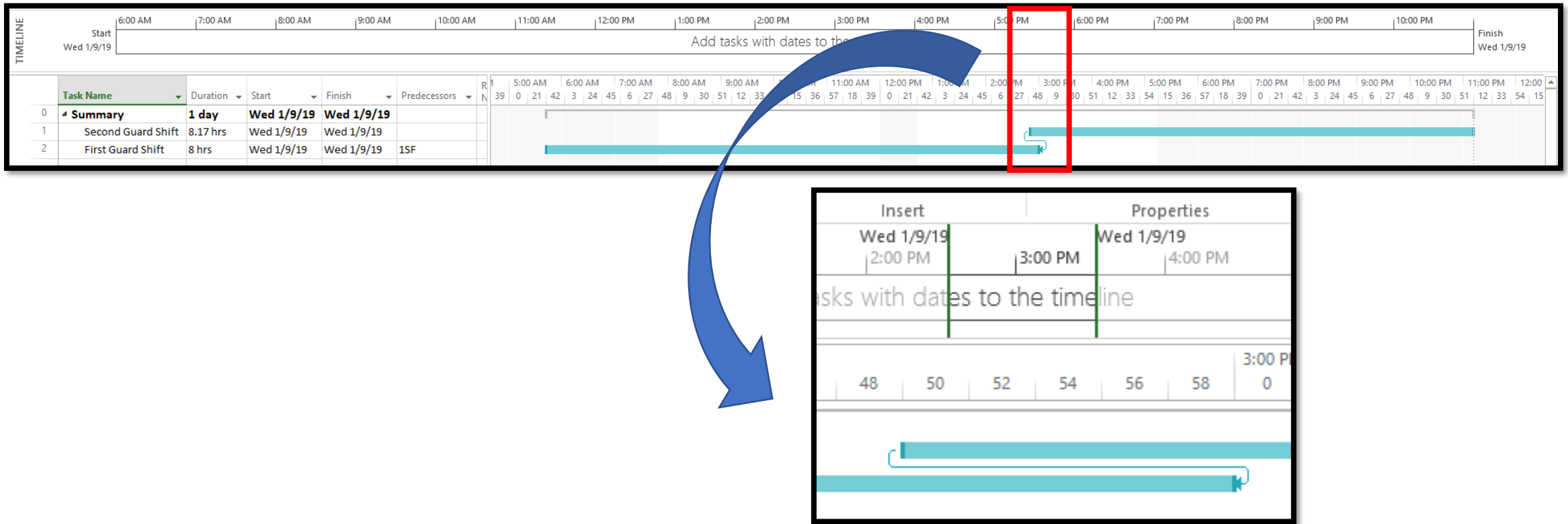
- Please make a note that the first guard shift person cannot stop working (finish) until the second guard shift person has begun (start) working.
- This relationship constricts when the predecessor can finish.
- Some industries (i.e., construction) do not use this relationship.

**Figure 6-11. Start-to-Finish (SF) Relationship**

# Start-to-Finish (SF) Relationship

## Example Two

- A security guard is working from 6 AM to 3 PM. This is the first guard shift.
- Another security guard shows up at 2:50 PM to start the second guard shift. There is a 10 minute overlap between the first and second shifts.
- The first shift guard can leave at any point after the second shift guard has started during this 10 minute overlap (SF relationship).



**Figure 6-12. Microsoft Project® Gantt Chart Start-to-Finish Relationship Example Two**

Above shows both guard shifts in their entirety.

Below shows a closeup of the 10 minute overlap between the first and second guard shifts.

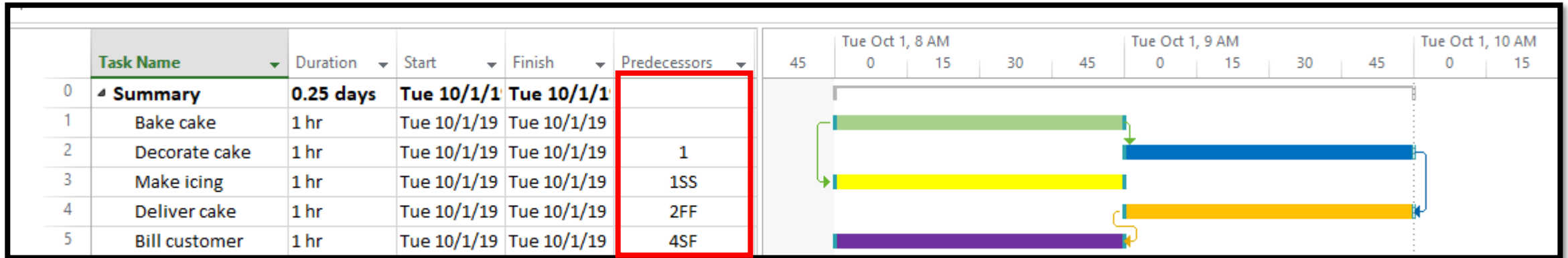
Note that in Microsoft Project® this Start-to-Finish dependency is connected to the first guard shift. The Second Guard Shift is a “predecessor” to the First Guard Shift in the sense that the Second Guard Shift must start (precede) before the First Guard Shift can finish.

# Making a Wedding Cake Example

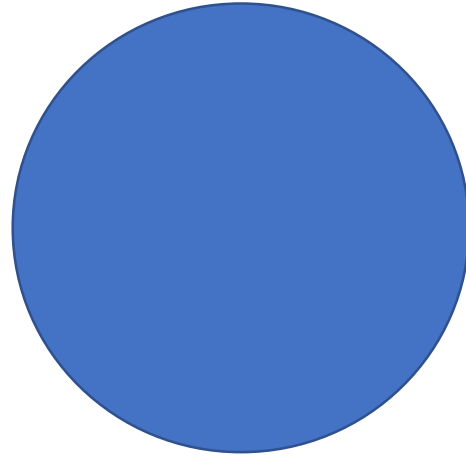
- Bake cake
- Decorate cake (Bake cake is a predecessor, FS)
- Make icing (Bake cake is a predecessor, SS)
- Deliver cake (Decorate cake is a predecessor, FF)
- Bill customer (Deliver cake is a predecessor, SF)

# Microsoft Project® Gantt Chart

## Making a Wedding Cake Relationship Example



**Figure 6-12A. Microsoft Project® Gantt Chart Making a Wedding Cake Relationship Example**



# Dependencies



# Project Dependencies Overview

- There are four project dependencies, and these can be used somewhat interchangeably.
- Which of these dependencies will be used in any specific project is determined through research (e.g., product analysis, contract analysis, and resource limitations).

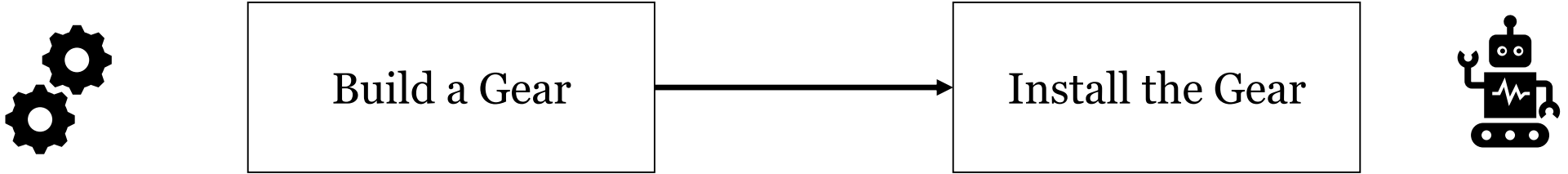
# Determining Project Dependencies

- Mandatory dependency
- Discretionary dependency
- External dependency
- Internal dependency

# Mandatory Dependency

- A mandatory dependency includes activities that must be completed before other activities can be started or finished.
- This dependency exists because of the physics of the situation or a contractual obligation.
- It is sometimes called hard logic.

# Mandatory Dependency Example



- We have to build the gear before we can install it.

**Figure 6-13. Mandatory Dependency Example**

# Discretionary Dependency

- Discretionary dependencies are just that, discretionary.
- Through practice, custom, or research, you and your project team learn the “best” way to execute an activity (best as in the quickest and/or cheapest way to complete the activity).

# Discretionary Dependency Flexibility

- In some cases (e.g., with soft-logic relationships), we do not have to execute an activity in the best way. That is, we have flexibility in how we approach an activity, and there are alternatives to the best way (e.g., at our discretion).
- Sometimes a less-than-optimum way must be selected because of resource constraints.

# Discretionary Dependency Labels

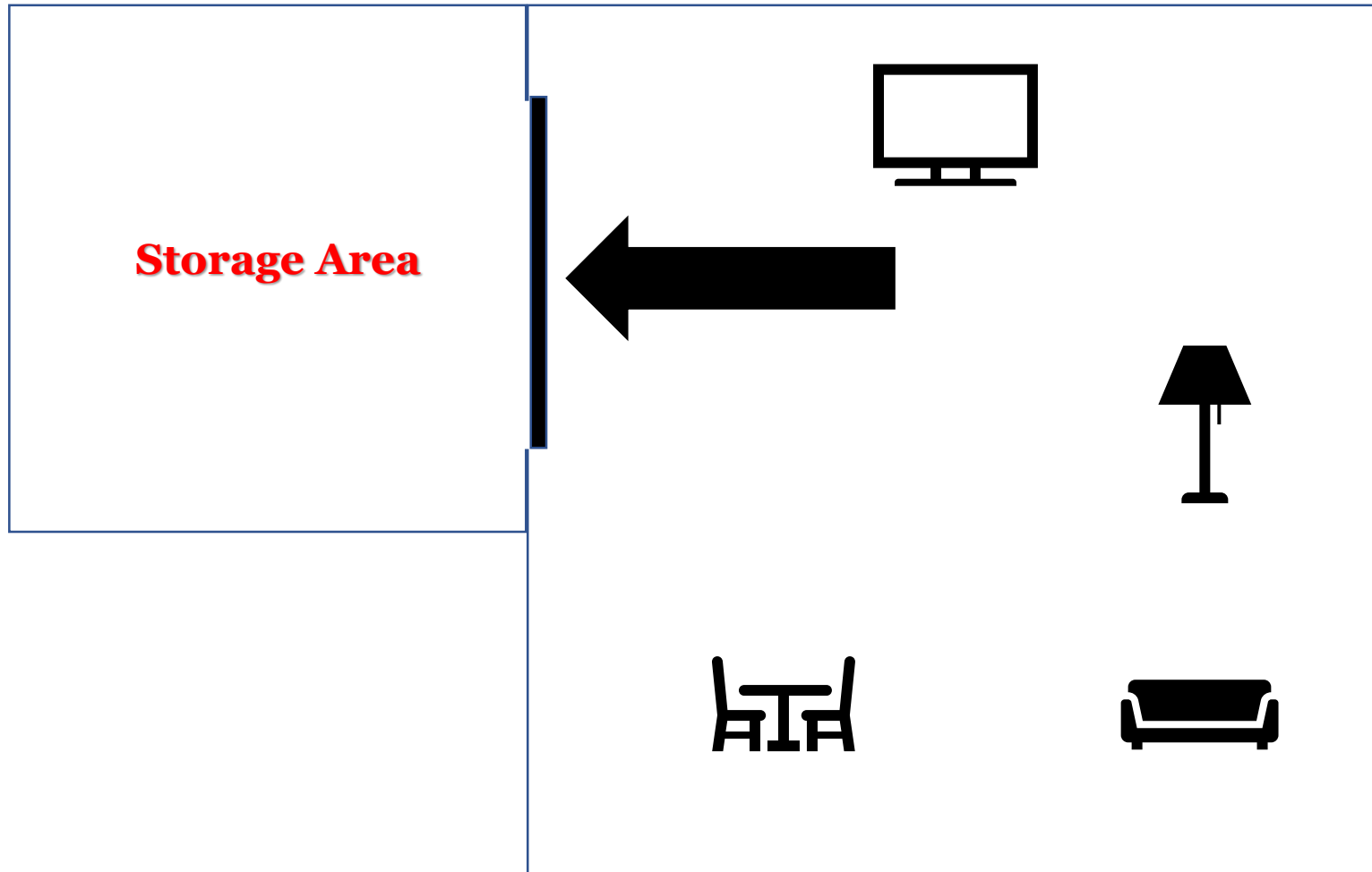
- The best approach is sometimes labeled as the preferred or preferential approach.
- The alternative approach is less than ideal (takes longer or is more expensive than the preferred approach).

# Discretionary Dependency Example

- In this example, we must wax an office floor.
- Part of the procedure involves moving office furniture off the floor before waxing the floor.
- Next, we will examine the two approaches to waxing the office floor.



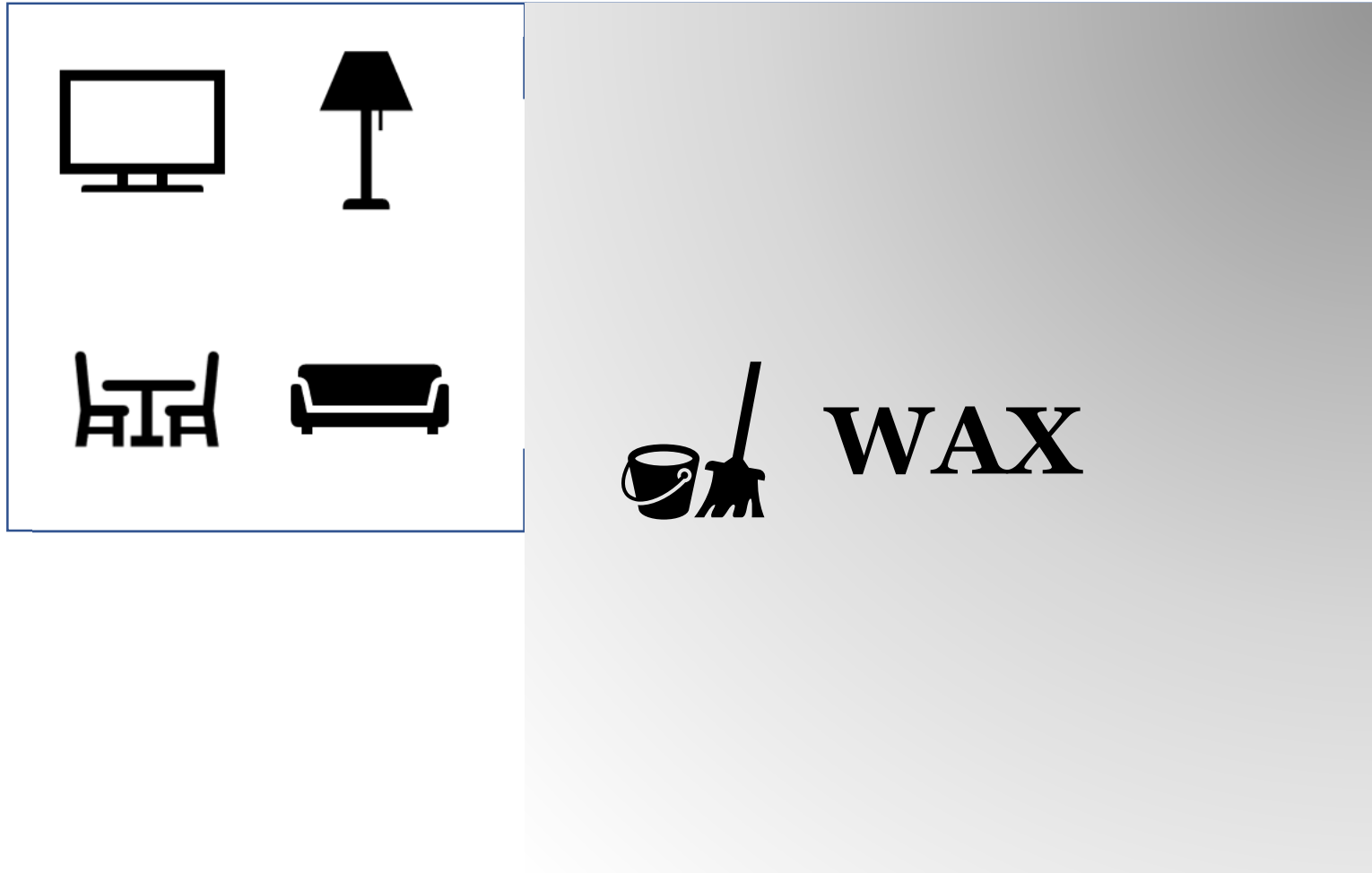
# Option 1: Preferred Approach



**Project: Wax the office floor.**

Step 1: Move the furniture from the office into the storage room (one hour).

**Figure 6-14. Discretionary Dependency Preferred Example**

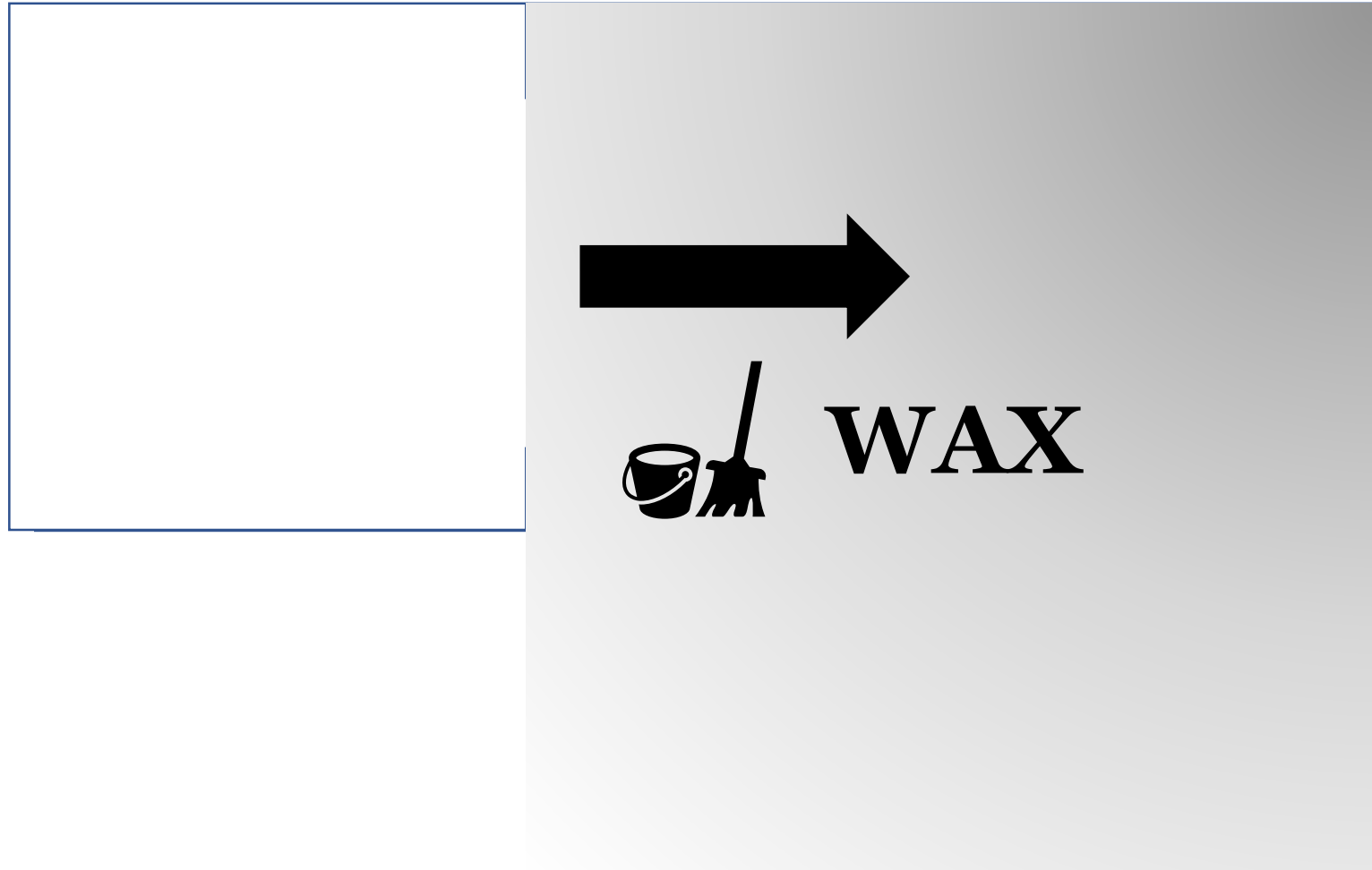


**Project: Wax the office floor.**

Step 2: Wax the office floor (one hour).

Step 3: Wait for the wax to dry (one hour).

**Figure 6-15. Discretionary Dependency Preferred Example**



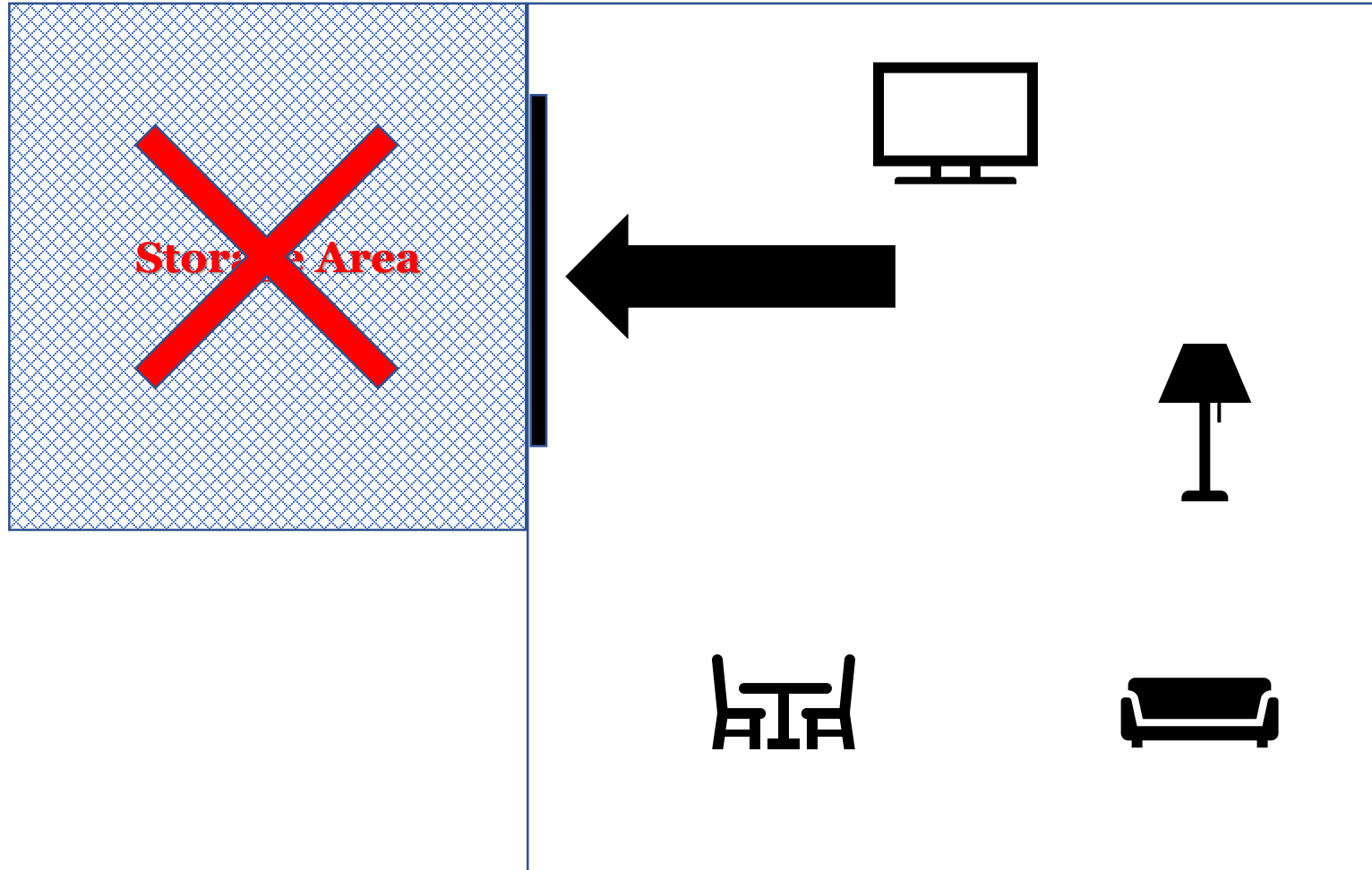
**Project: Wax the office floor.**

Step 4: Move the furniture back into the office (one hour).

Total time to complete the project: 4 hours.

**Figure 6-16. Discretionary Dependency Preferred Example**

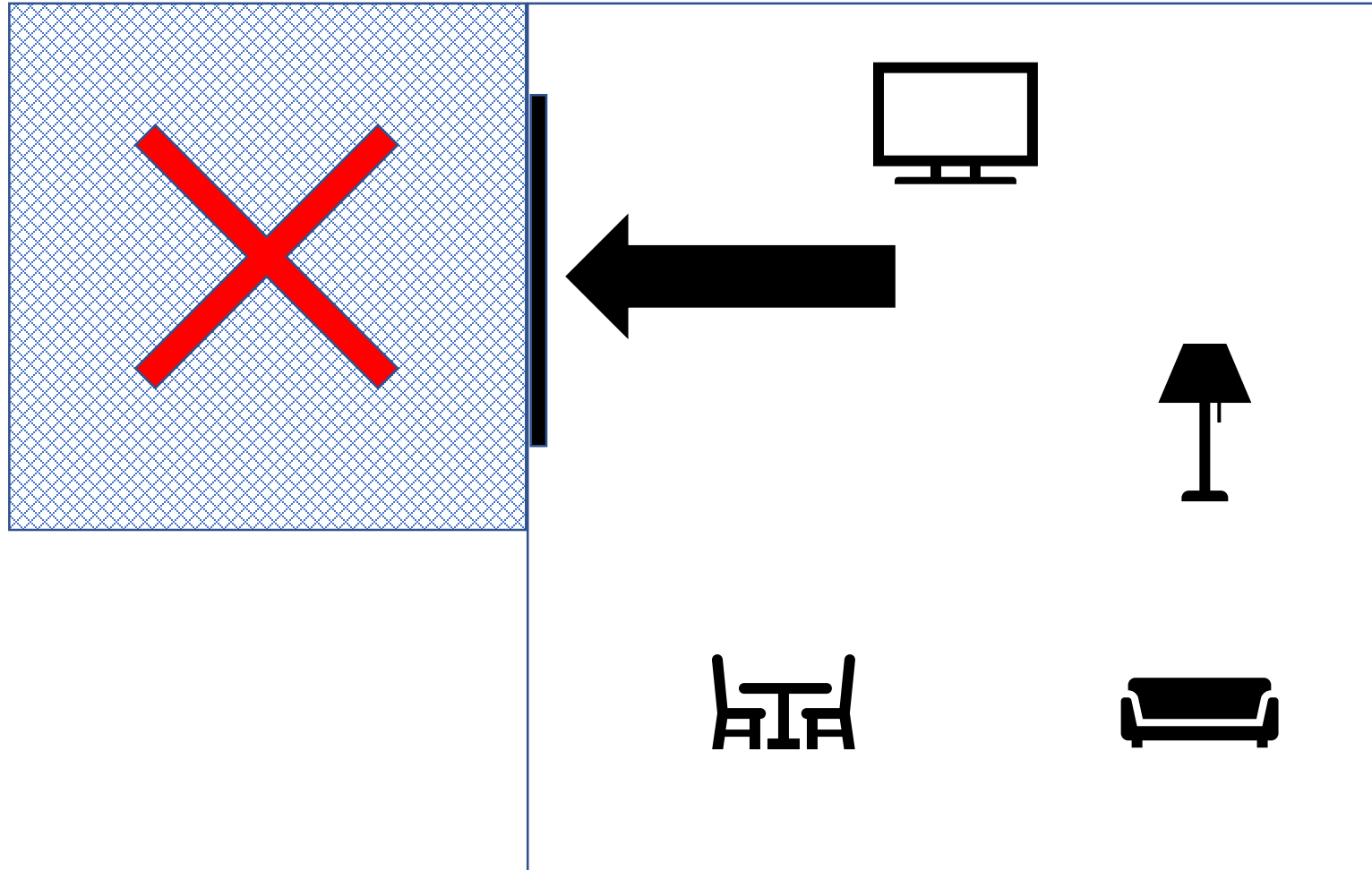
# Option 2: *Alternative Approach*



**Project: Wax the office floor.**

We need to wax the office floor, but the storage area is unavailable to store the office furniture. Therefore, we need to find an alternative to clearing the office floor for waxing.

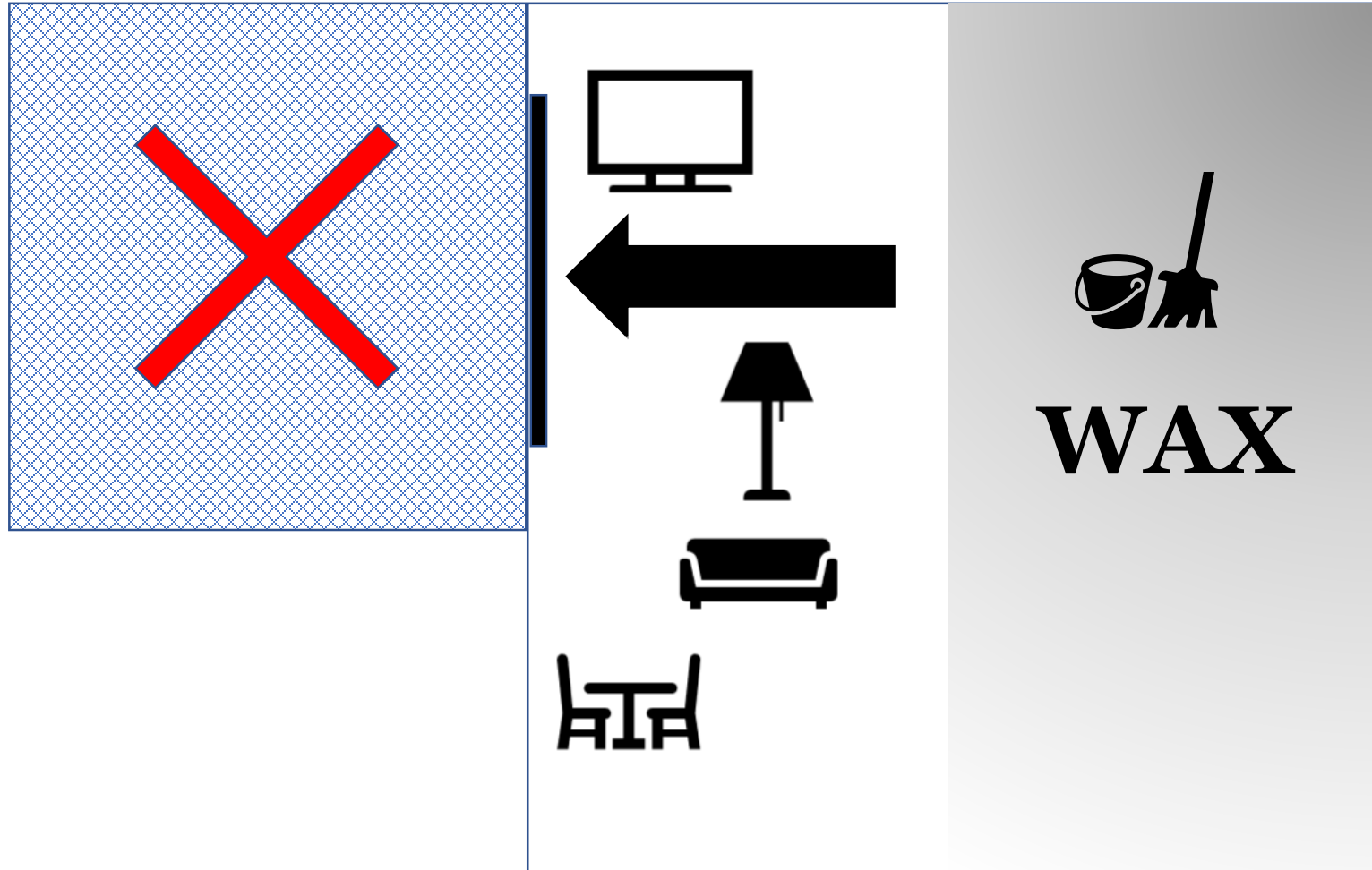
**Figure 6-17. Discretionary Dependency **Alternate** Example**



**Project: Wax the office floor.**

Step 1: Here is our alternative approach: move the furniture from one side of the office to the other side of the office (30 minutes).

**Figure 6-18. Discretionary Dependency **Alternate** Example**



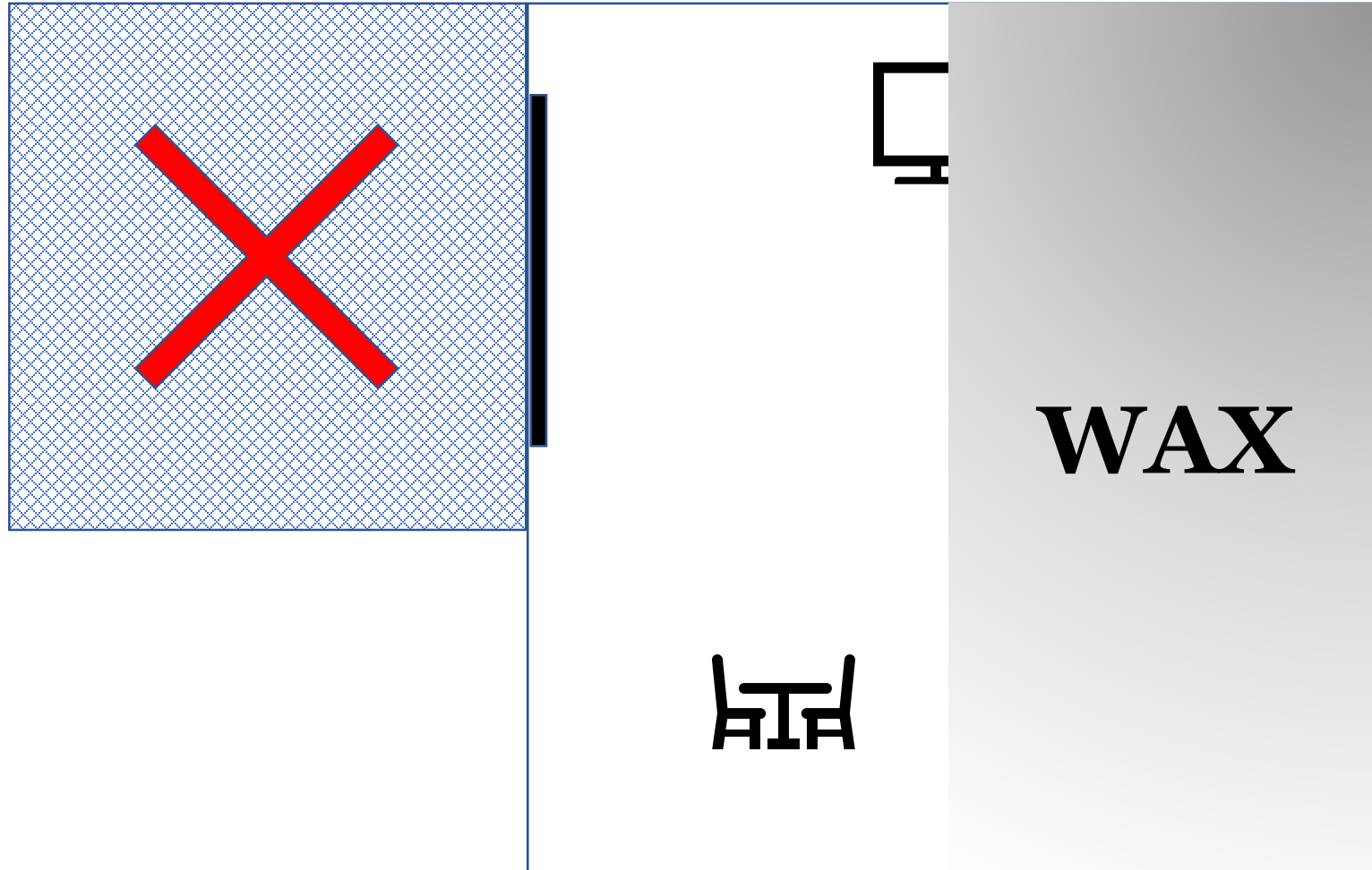
**Project: Wax the office floor.**

Step 2: Wax half the office floor (30 minutes).

Step 3: Wait for the wax to dry (one hour).

**Figure 6-19. Discretionary Dependency **Alternate** Example**

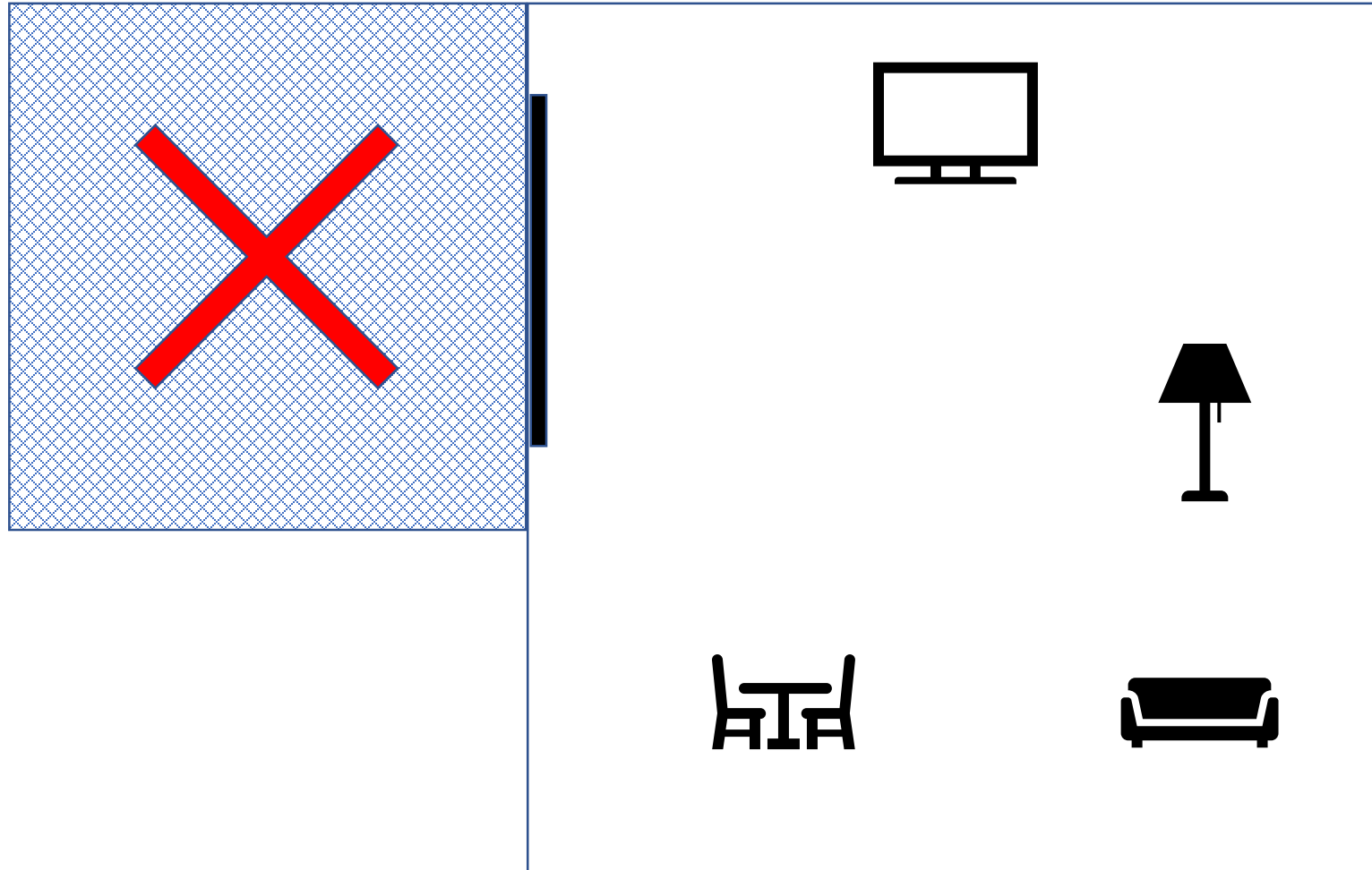




**Project: Wax the office floor.**

Step 4: Move the furniture from one side of the office to the other side of the office (30 minutes).

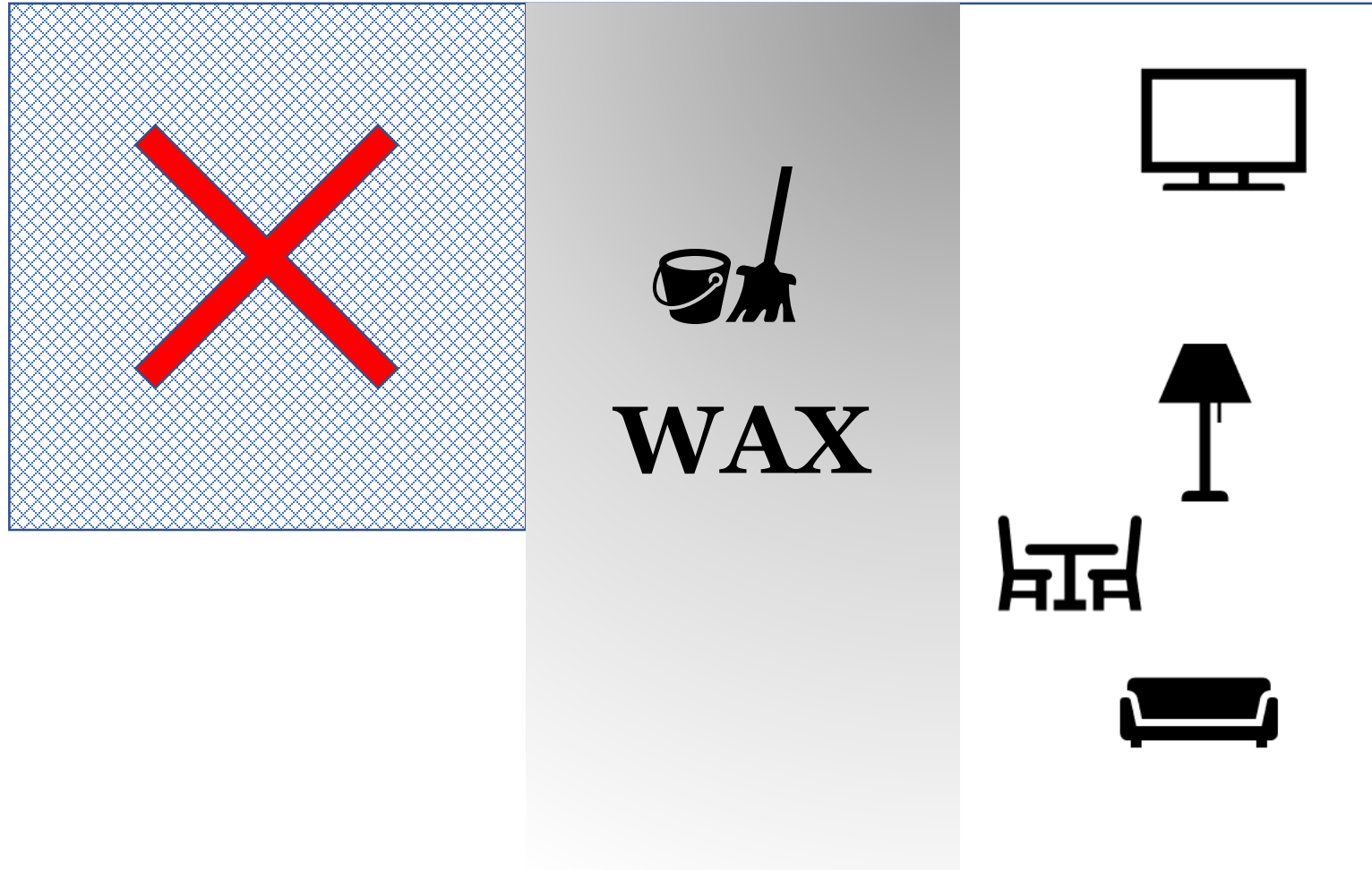
**Figure 6-20. Discretionary Dependency **Alternate** Example**



**Project: Wax the office floor.**

Step 4: Move the furniture from one side of the office to the other side of the office (30 minutes).

**Figure 6-20. Discretionary Dependency **Alternate** Example**

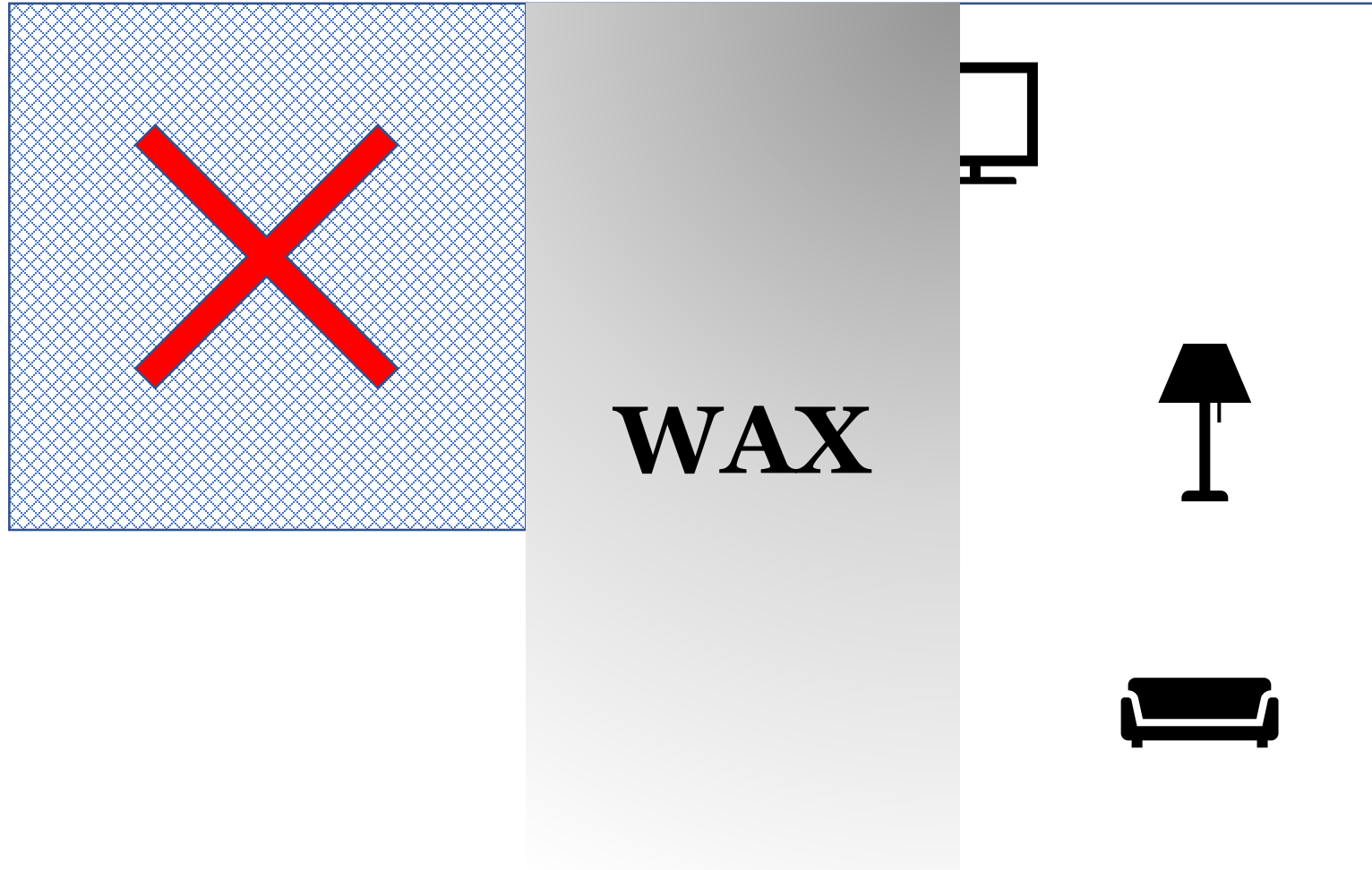


**Project: Wax the office floor.**

Step 5: Wax half the office floor (30 minutes).

Step 6: Wait for the wax to dry (one hour).

**Figure 6-21. Discretionary Dependency **Alternate** Example**



**Project: Wax the office floor.**

Step 7: Move the furniture back into its original arrangement on the office floor (30 minutes).

Total time to complete the project:  
4 hours and 30 minutes.

**Figure 6-22. Discretionary Dependency **Alternate** Example**

# Discretionary Example Comparisons

## Storage Area Available (Preferred)

- Step 1: Move the furniture from the office into the storage room (one hour).
- Step 2: Wax the office floor (one hour).
- Step 3: Wait for the wax to dry (one hour).
- Step 4: Move the furniture back into the office (one hour).
- Total time to complete the project: **4 hours**.

## Storage Area Unavailable

- Step 1: Here is our alternative approach: move the furniture from one side of the office to the other side of the office (30 minutes).
- Step 2: Wax half the office floor (30 minutes).
- Step 3: After the wax is dry (1 hour), move the furniture to the other side of the office (30 minutes).
- Step 4: Wax the other half of the office floor (30 minutes).
- Step 5: After the wax is dry (1 hour), move the furniture back into its original arrangement on the office floor (30 minutes).
- Total time to complete the project: **4 hours and 30 minutes**.

# Discretionary Example Conclusion

- The preferred approach (storage room available) is faster (4 hours).
- The alternative procedure is longer (4 hours and 30 minutes), but it does provide an option if the preferred approach is not available.

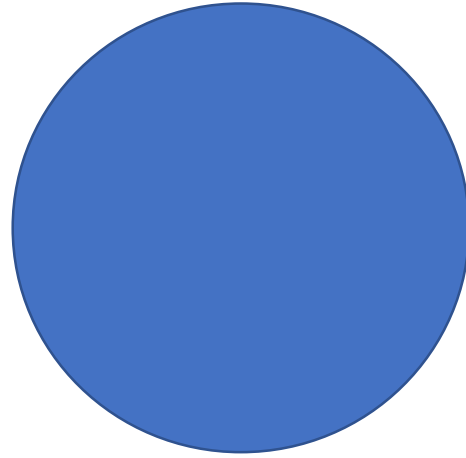
# External Dependency

- An external dependency includes activities tied to external non-project related activities.
- An external dependency could be mandatory or discretionary (as in a mandatory external or a discretionary external dependency).

# Internal Dependency

- An internal dependency is usually internal to the project and is within the project team's control.
- An internal dependency could be mandatory or discretionary (as in a mandatory internal or a discretionary internal dependency).





# Leads and Lags

# Lead

- A lead allows a change to a dependency that permits an acceleration of a dependent activity (the “to” as opposed to the “from” activity) in a relationship.
- Put another way, a successor activity can begin at the same time as its predecessor activity is still being executed (both activities are happening at the same time).
- Leads are only found in a finish-to-start relationship.

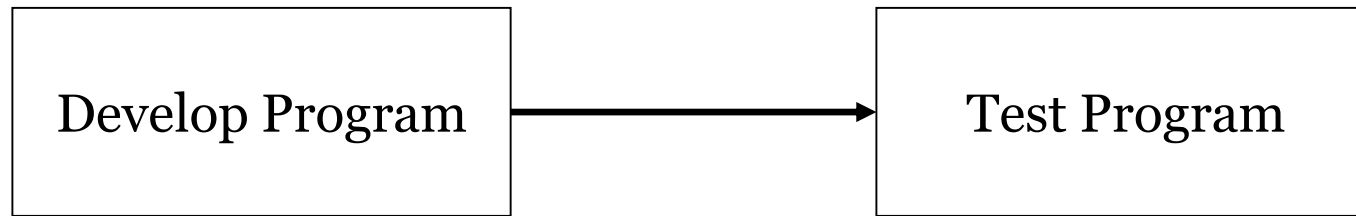
# Lag

- A lag causes a change to a dependency that increases time needed to complete a dependent activity.
- Put another way, lag is the delay of a successor activity. It represents a certain period of time that must pass before a successor activity can begin.
- No resources are consumed in a lag.
- Lags are found in all four relationship types.

# Lead Example

- In this example, there are two activities.
- In the first activity, a programmer is writing software that contains 1,000 lines of code. These 1,000 lines of code are divided equally into four modules (250 lines of code each).
- After the programmer writes the 1,000 lines of code, the software will be given to another programmer to test the 1,000 lines of code. The testing of the code is the second activity.

# Lead Example



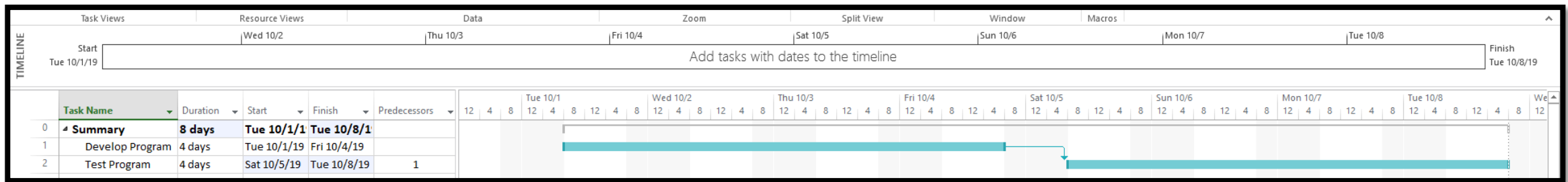
- **Lead (Finish-to-Start):** The Test (Software) Program activity cannot begin [start] until the Develop (Software) Program is completed [finish]. However, a part of the program (a module) is finished before the entire program is finished, and this module can be tested early. Time spent testing a module is lead time.

**Figure 6-24. Lead Example**

# Lead Example Before Lead Applied

- The Develop Program activity is four days long. It consists of developing four modules (one module per day). After the four modules are finished (that is, the Develop Program activity is finished), the completed program is an input into the Test Program activity.
- The Test Program activity is four days long (each module takes a day to test). Hence, the entire length of the project is eight days long.
- Note that this is a finish-to-start relationship between these two activities.

# Microsoft Project® Lead Example Before Lead Applied



**Figure 6-25. Microsoft Project® Lead Example Before Lead Applied**

# Lead Example After Lead Applied

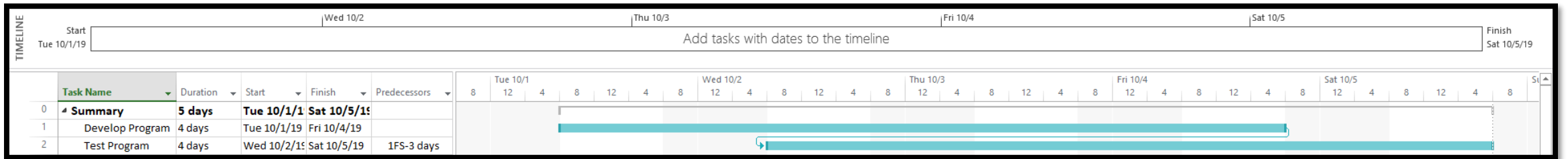
- After carefully examination of the Develop Program activity, we discover the four modules can be tested independently of each other. In other words, as each module is completed (each takes one day), it can be released to the Test Program activity.
- A three-day lead can be applied to the Test Program activity, which reflects the first module will be available as an input into the Test Program activity after one day.
- The entire project is shortened to just five days.



# Microsoft Project® Lead Note

- In Microsoft Project®, lead is treated (logically) as negative lag. Hence, when adding lead to a successor activity, it is entered as a negative number in the lag dialog box.
- In this example, three days of lead (-3) is added to the Develop Program activity.

# Microsoft Project® Lead Example After Lead Applied

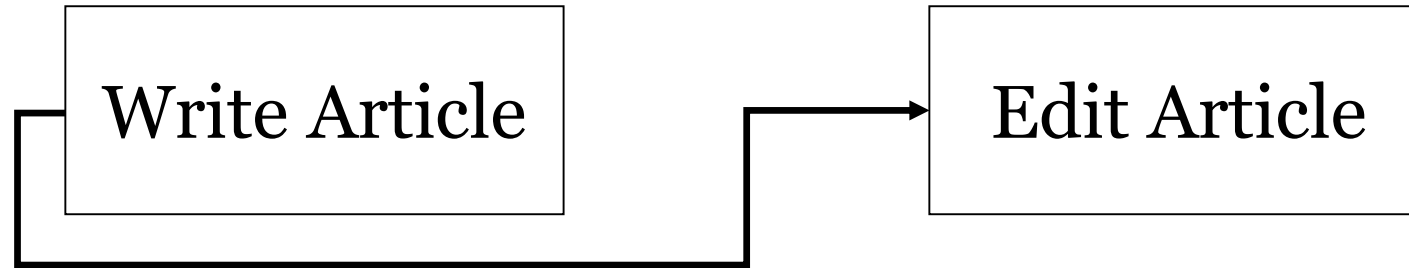


**Figure 6-26. Microsoft Project® Lead Example After Lead Applied**

# Lag Example

- In this example, there are two activities.
- In the first activity, an author is writing a newspaper article.
- After the author has written the newspaper article, an editor will review the article. The editing of the article is the second activity.

# Lag Example



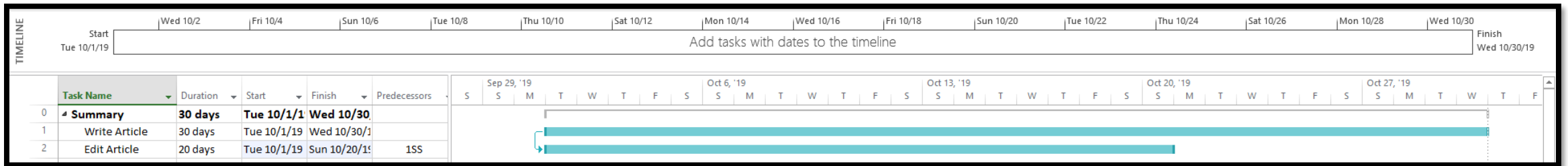
- Lag (Start-to-Start): The Edit Article activity cannot begin [start] until the Write Article activity has begun [start].

**Figure 6-27. Lag Example**

# Lag Example Before Lag Applied

- The Write (newspaper) Article activity is 30 days long. The relationship (start-to-start) suggests that immediately after the author begins to write an article, the newspaper editor can begin reviewing the article.
- The Editing Article activity takes 20 days. Since, both activities take place concurrently, the total project time is 30 days long.
- Note that this is a start-to-start relationship between these two activities.

# Microsoft Project® Lag Example Before Lag Applied

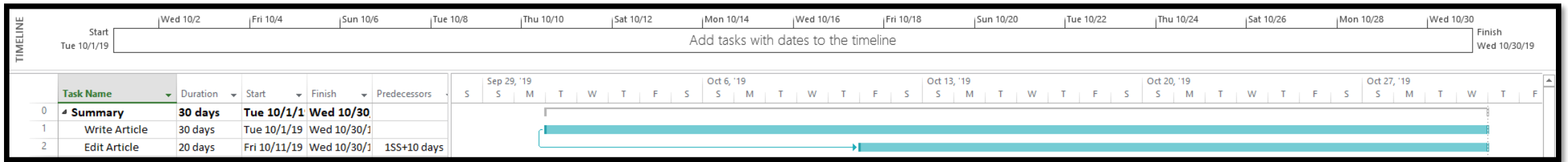


**Figure 6-28. Microsoft Project® Lag Example Before Lag Applied**

# Lag Example After Lag Applied

- As a practical matter, the author will not produce a reviewable article until 10 days into the writing-article process. Hence, a ten-day lag must be applied to the Edit Article activity. That is, the editor must let the author write for 10 days before beginning the editing process.
- The Editing Article activity takes 20 days. Adding ten days of lag to this activity does not increase the project length (it remains 30 days long).

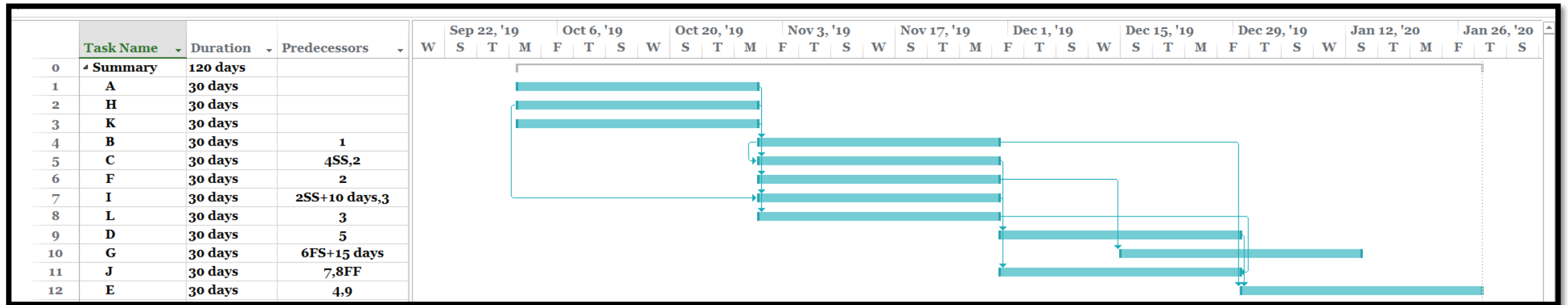
# Microsoft Project® Lag Example After Lag Applied



**Figure 6-29. Microsoft Project® Lag Example After Lag Applied**



# Microsoft Project® More Lag Examples



Lag is represented between Activity H and Activity I (SS + 10 days). That is, Activity I can begin no earlier than 10 days after Activity H has started. Lag is also represented between Activity F and Activity G (FS + 15 days). That is, Activity G cannot begin earlier than 15 days after Activity F has finished.

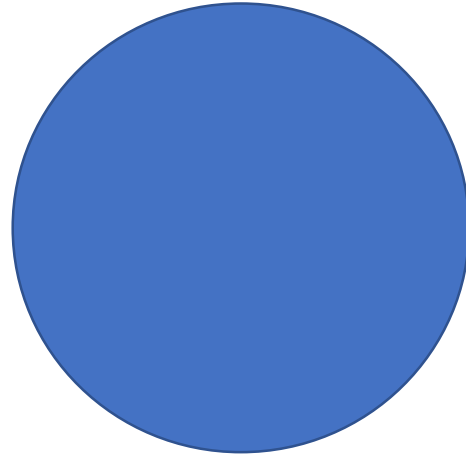
**Figure 6-30. Microsoft Project® More Lag Examples**

# Conditional Diagramming Methods (CDM)

- In some instances, precedence diagrams are not the best way to create a schedule.
- If a schedule has non-sequential looping and conditional branching (as in software development), a CDM may be better.
- A CDM allows for activities that may need to be repeated.
- Two popular CDMs are Systems Dynamics and Graphical Evaluation and Review Technique (GERT).

# Schedule Network Templates

- Although projects produce unique deliverables, some of the scheduling processes may be similar.
- Templates can be used when parts of projects contain duplicate information or when similar scheduling requirements exist between projects.
- Reuse of diagrams through templates can significantly reduce schedule planning preparation.
- A portion of a schedule network diagram is sometimes called a subnetwork or fragmented network.



# Durations

# Estimating Activity Durations Overview

- To this point, we have identified and sequenced activities.
- In order to complete a schedule, we must also estimate how long each activity will take.
- Again, if the data are less precise or of inferior quality, the quality of duration estimates will likewise be less precise (e.g., larger ranges).

# Activity Duration Considerations

- Improvements in technology may improve productivity.
- Other important factors to consider when analyzing activity durations include the law of diminishing returns and staff motivation.

# Law of Diminishing Returns and Resources

- As resources are added, returns diminish. At some point, returns become negative.
- Adding resources does not necessarily add to productivity in a linear way.

# Staff Motivation

- *Student syndrome*: Student syndrome describes a situation in which people wait to the last possible moment to complete tasks (procrastination) (Smith, 2010).
- *Parkinson's law*: Parkinson's law describes a situation in which people expand work to fit the available time (Parkinson, 1955).



# Activity Duration Estimation Methods

- If an activity is relatively simple and well understood and we have a great deal of experience executing this activity, calculating duration may be relatively easy. In fact, calculation in this sense may simply be the anticipated duration (e.g., based on previous experience).
- On the other hand, an activity may be complex and require a more analytically rigorous approach to calculating its duration. We will concentrate on the latter.

# Bottom-Up Estimating

- If upon initial planning an activity's duration cannot be sufficiently estimated, it can be broken down into smaller, more detailed, fundamental parts.

# Alternatives Analysis

- Activities can be completed using several methods. Selecting the best alternatives for accomplishing scheduled activities is an important part of the planning process.
- Some parameters to be considered in selection include personnel skill level, equipment capability and manufacturer, and in-house development versus contractor production (make-or-buy decisions).
- When exploring alternative ways for accomplishing an activity, we always want to examine the cheapest (i.e., cost) way to accomplish an activity and still complete it on time. This does not mean we will adopt the cheapest alternative, but we want to at least explore this option.

# Top-Down Estimation Methods

- All things being equal, top-down approaches are less time consuming and less costly to produce, but they also tend to be less accurate than other methods.
- Top-down estimates are typically used earlier in a project or in instances where activities are difficult to analyze.
- Popular top-down methods include analogous and parametric estimations.

# Analogous Estimating

- Analogous estimates use experience from previous studies to calculate estimates for current or future projects.
- Subject matter experts are often engaged to calculate a project duration when using this methodology.
- Analogous estimates are most useful when the current or future project is similar in scope, cost, and so on to the reference project.

# Parametric Estimating

- “... [U]ses validated relationships between a project's known technical, programmatic, and cost characteristics and known historical resources consumed during the development, manufacture, and/or modification of an end item” (Joint Industry/Government Parametric Cost Estimating Initiative, 1999, p. 1-1).

# Three-Point Estimation

- A more quantitative approach to calculating an activity duration is the three-point estimation technique. Information generated from the three-point method can be applied, in turn, to calculate the overall duration of a project using the Program Evaluation and Review Technique or PERT (Johnson, Kotz, & Balakrishnan, 1995).
- A three-point estimation is especially useful when there is a substantial element of ambiguity in cost or duration estimates.

# Three-Point Estimating Technique

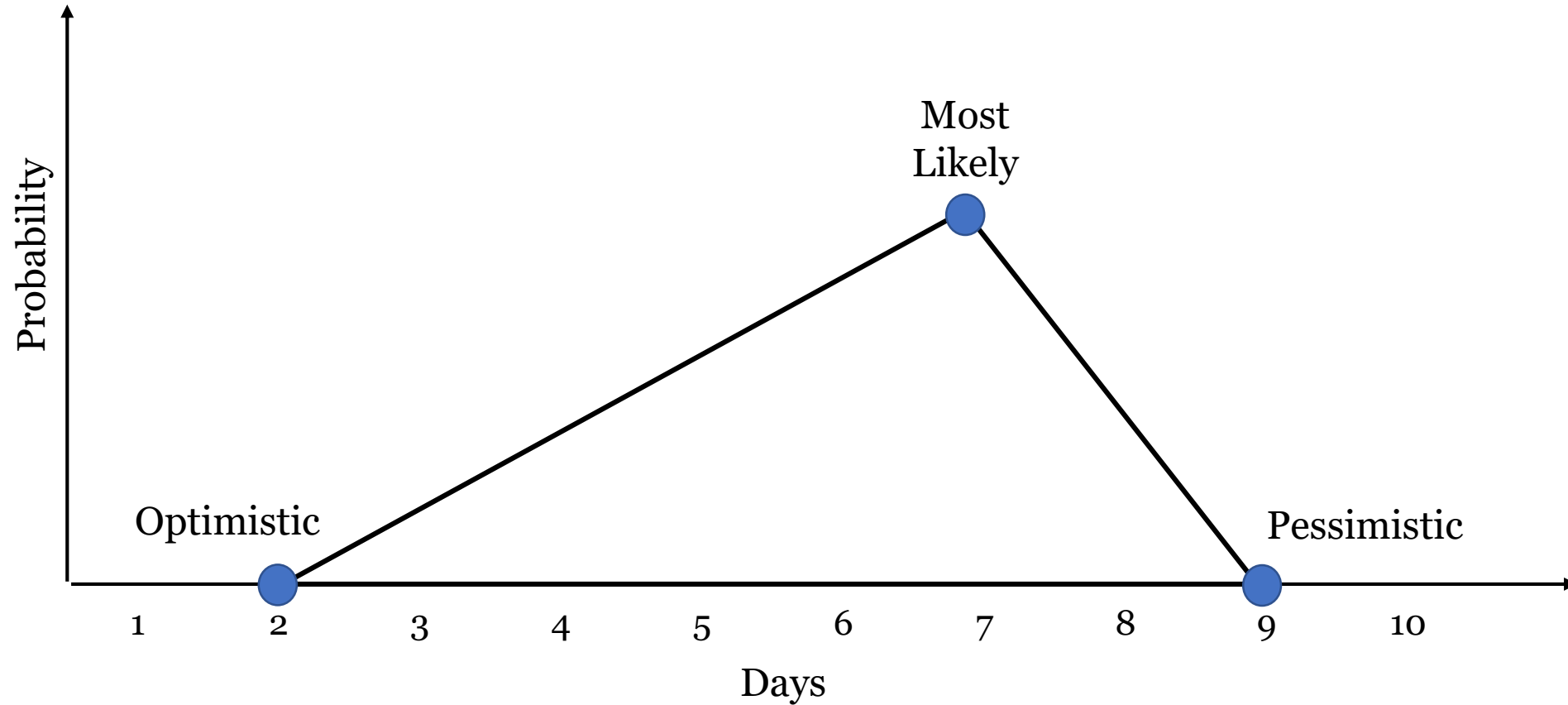
- To calculate a three-point estimate for an activity, we examine the range of possibilities for that activity and determine the optimistic (low), pessimistic (high), and most likely durations for that activity.



# Three-Point Estimating: Triangular Distribution

- If we believe the distribution of the optimistic, pessimistic, and most likely durations for an activity follows a triangular distribution, we employ the triangular estimation three-point formula.
- The triangular distribution is a simple average.

# Triangular Distribution Representation



**Figure 6-31. Triangular Distribution Representation**

# Expected Value Triangular Distribution

- $tE = \frac{(tO + tP + tM)}{3}$

- Where:

- $tO$  = optimistic estimate
- $tP$  = pessimistic estimate
- $tM$  = most likely estimate

- Note  $t$  is for time;  $c$  is for cost. We are using time information in this example.

# Standard Deviation for a Triangular Distribution

$$\bullet \sigma = \sqrt{\frac{((P - O)^2 + (M - P)(M - O))}{18}}$$

- Where:
  - $tO$  = optimistic estimate
  - $tP$  = pessimistic estimate
  - $tM$  = most likely estimate

**A standard deviation question on a triangular distribution is not likely to be on the PMP Exam.**

# Three-Point Estimating: Beta Distribution

- If we believe the distribution of the optimistic, pessimistic, and most likely durations for an activity follows a normal distribution, we employ the beta distribution three-point estimation formula.
- The beta distribution is derived from the program evaluation and review technique (PERT). It is sometimes called the PERT estimate. It is a weighted average.

# PERT\* Estimate

- $$tE = \frac{(tO + tP + (4 \times tM))}{6}$$

- Where:

- $tO$  = optimistic estimate
- $tP$  = pessimistic estimate
- $tM$  = most likely estimate

\*Program Evaluation and Review Technique

# PERT\* Estimate Example

- Note the following calculations for an activity:
- $o = 2$  days,  $p = 7$  days,  $m = 5$  days.
- $(2 + 7 + (4 \times 5)) \div 6 = 4.83$  days.
- The expected value for this activity is 4.83 days.

\*Program Evaluation and Review Technique

# PERT\* Estimate Example Standard Deviation

- From the previous example, it is possible to approximate the standard deviation or sigma ( $\sigma$ ) of the expected value:
- One  $\sigma$  for a beta distribution equals  $(p - o) \div 6$ .
- In this example, one  $\sigma$  equals  $(7 - 2) \div 6 = .83$  days.
- Note that the standard deviation probabilities are assumed to follow a normal distribution.

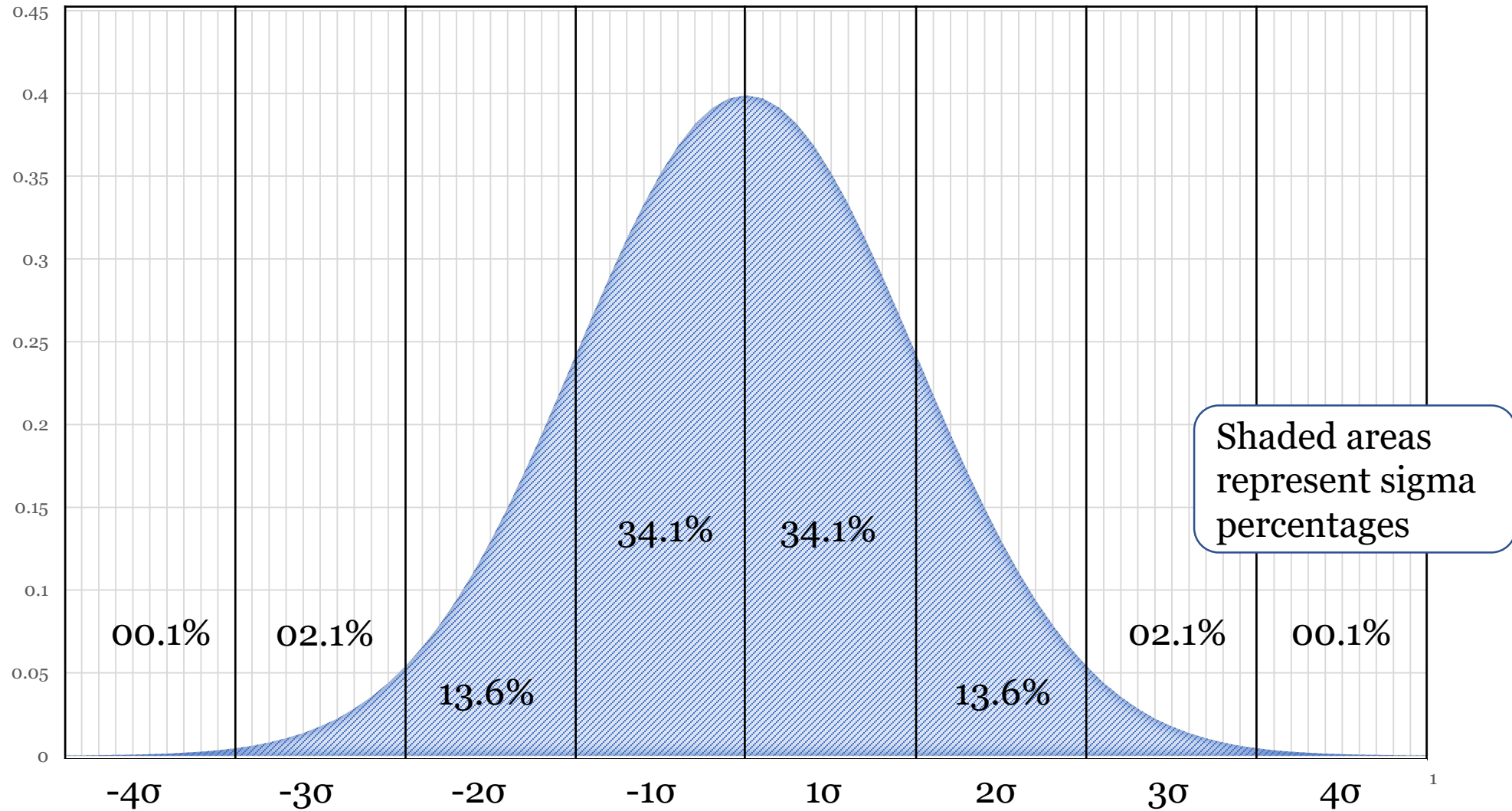
\*Program Evaluation and Review Technique



# Normal Distribution and Sigma

- One standard deviation equals 68.26%
- Two standard deviations equals 95.44%
- Three standard deviations equals 99.73%

# Normal Distribution Plot



**Figure 6-32. Normal Distribution Plot**

# Standard Deviation for PERT\* Estimate Ranges

- From the previous example,  $1 \sigma = 4.83 \pm .83$  days, which means 68.26% of the time the activity will be completed between **4 and 5.67 days.**
- $2 \sigma = 4.83 \pm 1.67$  days; 95.44% completed between **3.17 and 6.5 days.**
- $3 \sigma = 4.83 \pm 2.5$  days; 99.73% completed between **2.33 and 7.33 days.**

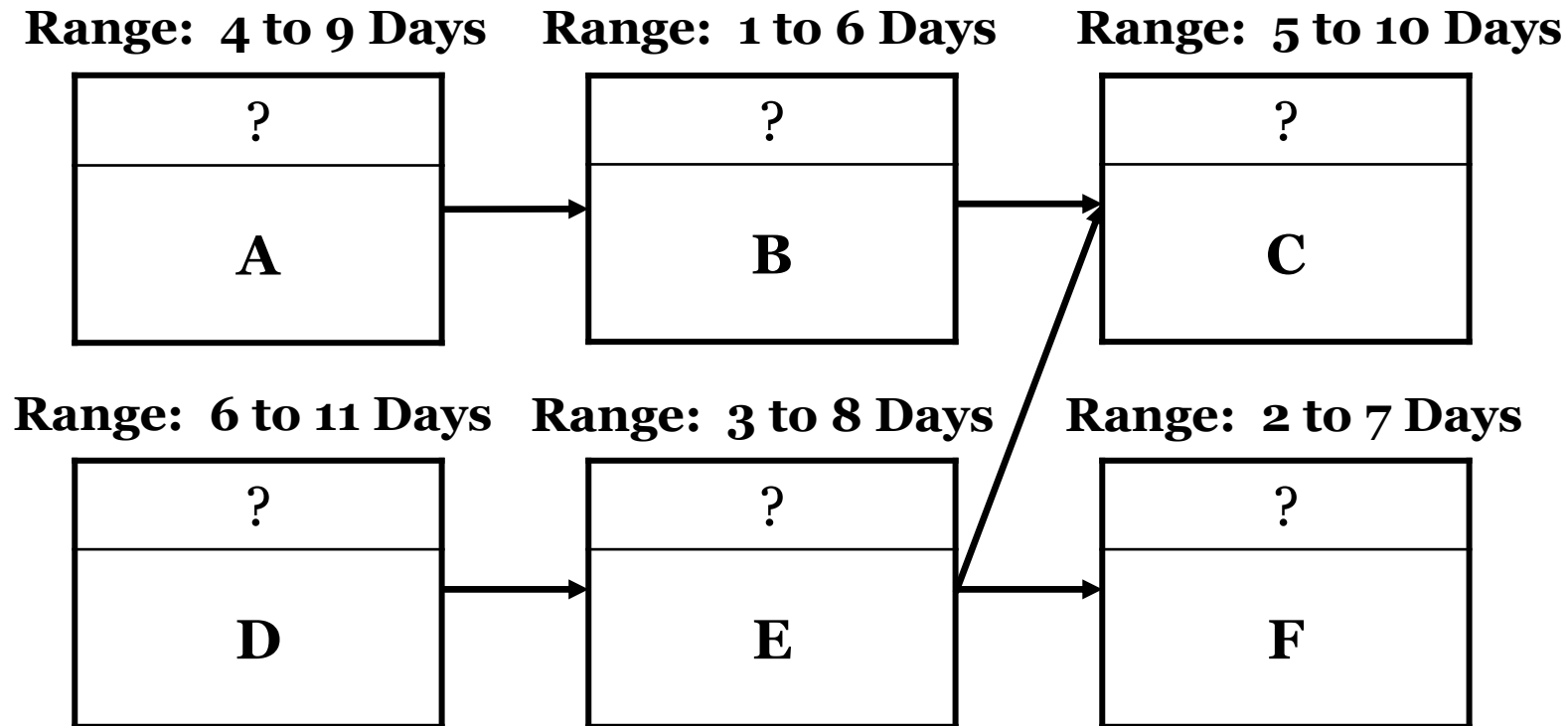
\*Program Evaluation and Review Technique

# Project-Wide PERT\*

- PERT can be used to model all activities and tasks in a project. It was developed by Booz Allen Hamilton in 1958 to assist the Navy in the development of the Polaris missile.
- Among its many functions, PERT provides a standard deviation for the summation of all activities within a given time period.
- PERT uses the Beta distribution when calculating estimates.

\*Program Evaluation and Review Technique

# PERT\* Analysis Example Diagram



**Figure 6-33: PERT Analysis Example: Range of Duration Possibilities for Each Activity**

\*Program Evaluation and Review Technique

# PERT\* Analysis Example

- We will conduct a PERT analysis on the critical pathway (D-E-C) using the low, high, and most likely estimates.
- We will assume the range of possible dates for each activity follows a normal distribution. Hence, it follows that an approximation of the most likely estimate for each activity is the mean of the range of those possibilities. For example, Activity A can be 4, 5, 6, 7, 8, or 9 days long. We will assume the mean of the sum of these possibilities (6.5 days) is the most likely estimate.

\*Program Evaluation and Review Technique

# PERT\* on the Critical Pathway Example

Model Information						
Activity	Low (o)	Most Likely (m)	High (p)	EV $(o + (4 \times m) + p) / 6$	SD $(p - o) / 6$	SD <sup>2</sup>
D	6	6.5	11	7.17	0.8333	0.6944
E	3	5.5	8	5.50	0.8333	.0.6944
C	5	7.5	10	7.50	0.8333	0.6944
Sums:				<b>20.17</b>		<b>2.0833</b>
Square Root of Sum:						<b>1.4434</b>

**Table 6-2. Program Evaluation and Review Technique Analysis on the Critical Pathway**

\*Program Evaluation and Review Technique **Example**

# Project-Wide PERT\* Standard Deviation

- Sum of the standard deviations squared:  $\sum s^2 = 2.0833$
- Take the square root of that value:  $\sqrt{2.0833} = 1.4434$
- One standard deviation (total across all activities) = **1.4434**

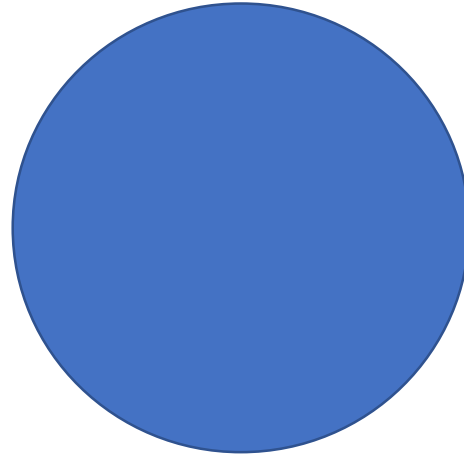
\*Program Evaluation and Review Technique



# PERT\* Analysis Results

- The expected value for the critical pathway is 20.17 days. One standard deviation for the critical pathway is 1.4434
- Probability the project will be complete in these ranges (standard deviations):
  - 68.26% chance it will be complete between 18.72 to 21.61 days
  - 95.44 % chance it will be complete between 17.28 to 23.05 days
  - 99.73% chance it will be complete between 15.84 to 24.50 days

\*Program Evaluation and Review Technique



# Schedule

# Creating the Schedule

- We now have all the elements we need to create the schedule.
- We have identified the activities.
- We know what resources will be needed to complete each activity.
- We know what sequence the activities will be performed in.
- Finally, we know how long each activity will take to perform.
- With this information, we can create a schedule.

# Schedule Development Methods

- Powerful schedule development methods include schedule network and critical pathway analysis, resource optimization, and Monte Carlo analysis.

# Schedule Network Analysis

- The purpose of schedule network analysis is to calculate the critical pathway(s) of a schedule and assist in critical chain project management (Goldratt, 1997).
- Resource smoothing and leveling are also important elements of schedule network analysis.
- Another dimension of schedule network analysis is what-if analysis.

# Schedule Network Analysis

- Schedule network analysis is achieved through calculating early starts and early finishes as well as late starts and late finishes for each activity.
- This information also helps us calculate float (slack).

# Critical Path Method

- A critical pathway is the most important pathway in a schedule because it is the longest.
- By definition, therefore, any time that is added to a critical pathway will necessarily make a project longer.
- For that reason, we must closely manage activities that are on a critical pathway (e.g., prevent bottlenecks).
- There may be more than one critical pathway (two or more pathways that are tied for the longest pathway).

# Critical Path and Float

- Again, a critical pathway is the longest pathway in a schedule.
- In addition, each activity on a critical pathway will have zero total float (slack). More about this later.



# Near-Critical Path

- A near-critical path is any pathway that is close in duration (near zero float) to the critical path.
- Note also that a project can have multiple critical pathways (that is, more than one pathway tied for the longest duration).
- Projects that have multiple critical pathways or near-critical pathways are harder to manage because multiple potential bottlenecks will have to be managed.

# Schedule Network Analysis Calculations for a Critical Pathway

- Forward pass
- Backward pass
- Total float (slack)
- Free float (slack)

# Forward Pass (Early Start or ES)

- Conducting a forward pass allows us to calculate the early start and early finish dates for each activity.
- To begin, consult the schedule management plan and note the date that the earliest activity\* is scheduled to start (day one).
- Generally speaking, any activity that has no predecessors and is not limited by resource constraints, usually starts on the first day of the project (day one).

\*More than one activity may begin on the start date.

# Forward Pass (Early Finish or EF)

- Once you have calculated the early start date for the first activity, calculate the earliest finish date for that activity.
- For example, an activity that consists of three consecutive day shifts\*\* would begin on day 1 at 8 AM and finish on day 3 at 5 PM.
- Hence, in this example, early start is day 1 and early finish is day 3.

\*\*Eight-hour work periods from 8 AM to noon and 1 PM to 5 PM each day (one hour off for lunch). This is a standard default work day in Microsoft Project®.

# Forward Pass (cascade forward)

- After calculating the early finish for the first activity in the project, cascade (move) forward to the next activity in a sequence.
- In our previous example, the activity finishes on day 3 (5 PM). Therefore, any activities that follow a sequence from the first activity could start as early as day 4 (8 AM).
- If the second activity in a sequence is four days long, its early start would be day 4 (8 AM) and its early finish would be day 7 (5 PM).
- Continue to cascade forward in this manner to the next activity in the schedule.

# Forward Pass (First Pass)

- A forward pass is sometimes called a *first pass*.

# Forward Pass Placement in Node

- In a precedence diagram node (activity), the early start often appears in the upper left-hand part of the node; the early finish often appears in the upper right-hand part of that node.\*

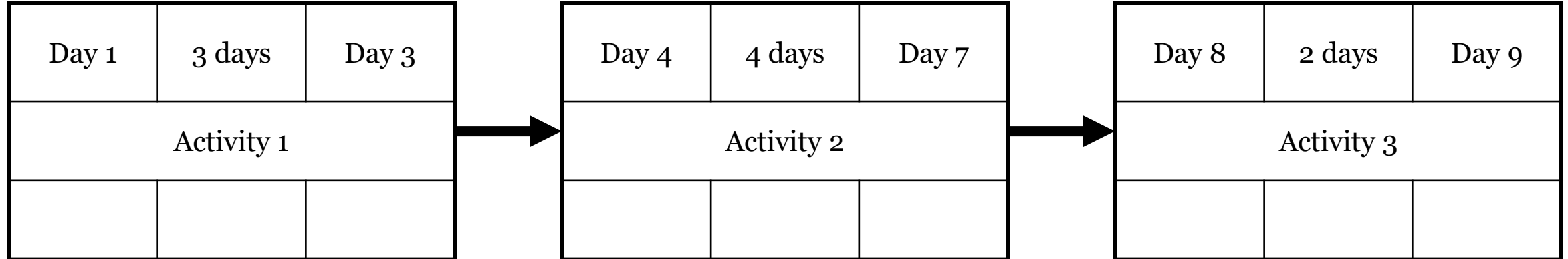
<b>Early Start</b>	Duration	<b>Early Finish</b>
Activity Name		

\*There is no standard for where information is placed on a node or what information appears on a node.

**Figure 6-34: Forward Pass Placement in Node**

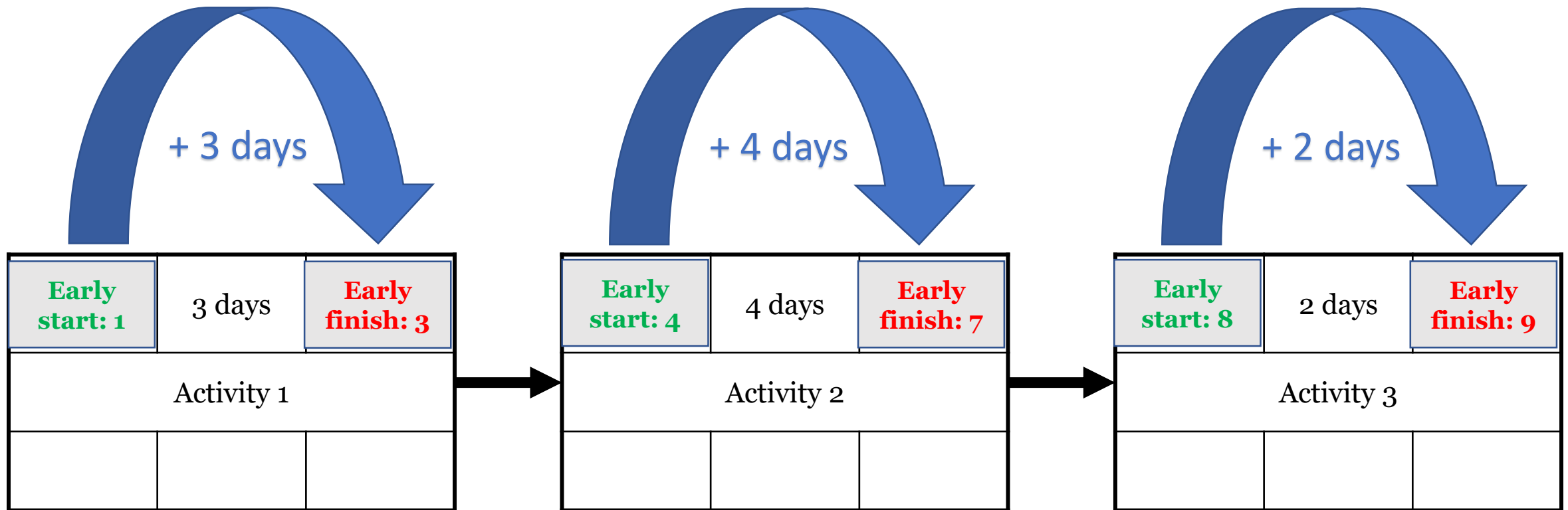
# Forward Pass Cascading Example One

- Note the early start and early finishes for each node.

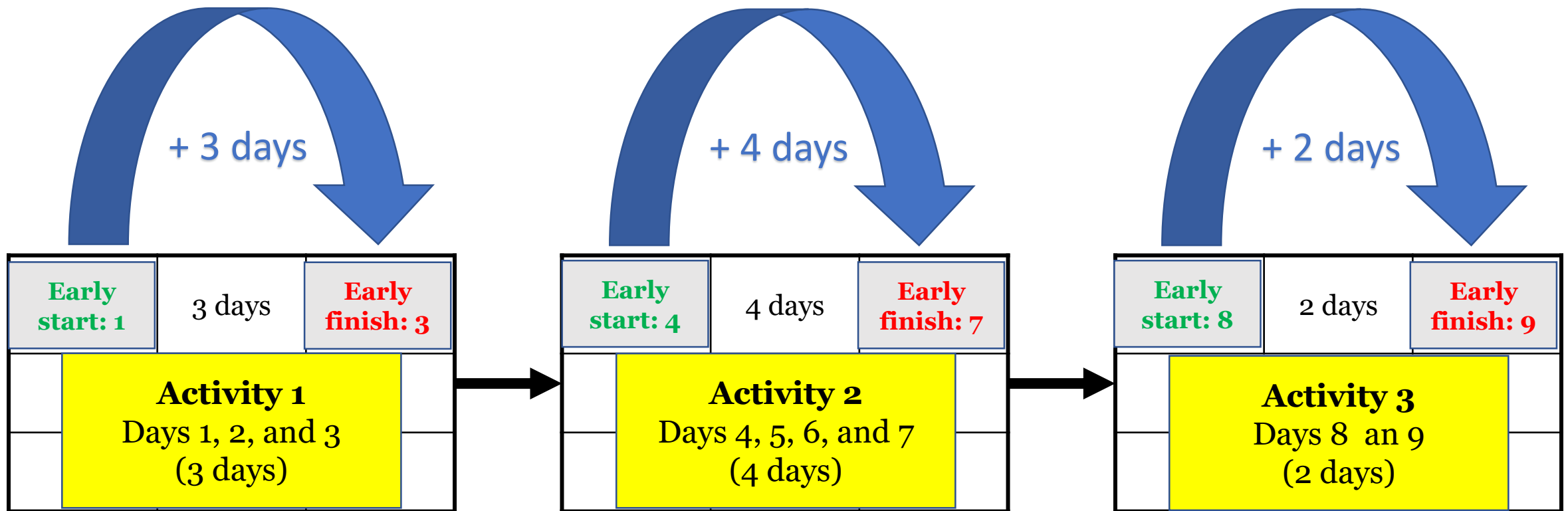


**Figure 6-35A: Forward Pass Cascading Example One**



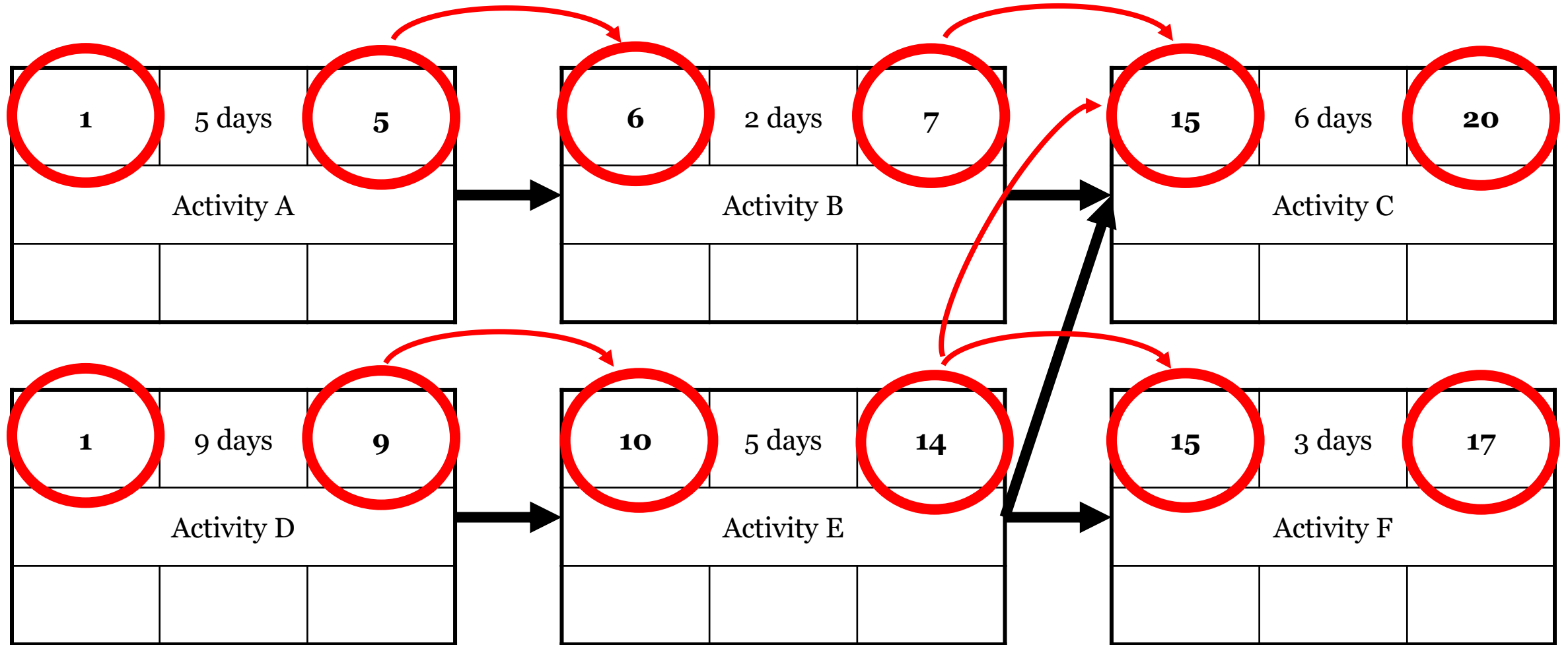


**Figure 6-35B: Forward Pass Cascading Example One**



**Figure 6-35C: Forward Pass Cascading Example One**

# Forward Pass Cascading Example Two



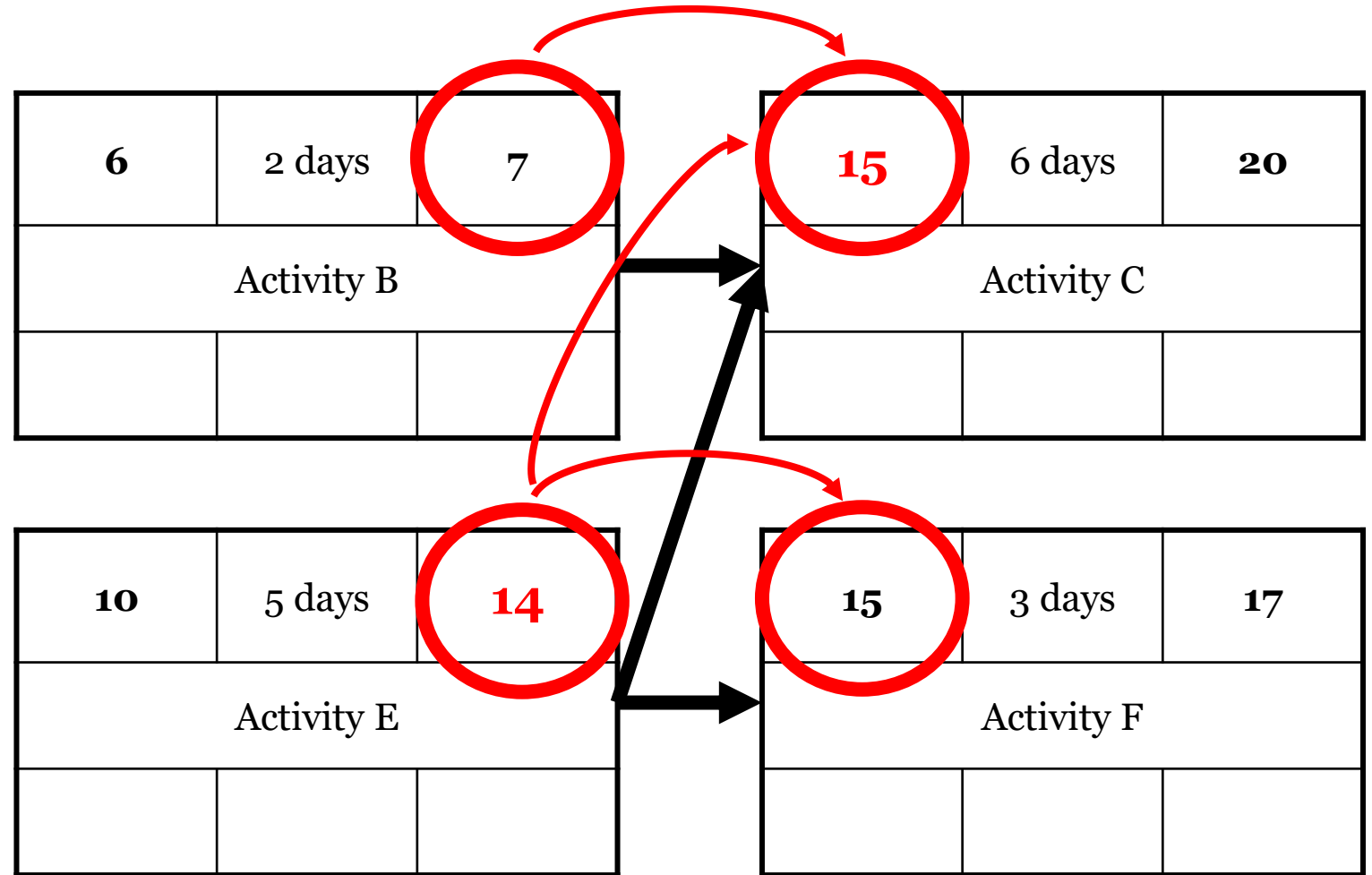
**Figure 6-36: Forward Pass Cascading Example Two**

# Forward Pass Dependencies

Note that the early start for Activity C is day 15. This is because Activity C has two dependencies, Activity B and Activity E.

In the event that an activity has more than one dependency, all predecessor activities have to be complete before the successor activity can begin.

Hence, both Activity B (early finish day 7) and Activity E (early finish day 14) have to finish before Activity C can begin (early start day 15).



**Figure 6-37: Forward Pass Dependencies**

# Backward Pass (Late Finish or LF)

- Conducting a backward pass allows us to calculate the late start and late finish dates for each activity.
- To begin, consult the schedule management plan and note the date that the project must finish on. This date is the late finish date for the last activity\* in the project.

\*More than one activity may finish on the late finish date.

# Backward Pass (Late Start or LS)

- Once you have calculated the late finish date for the last activity, calculate the late start date for that activity.
- For example, suppose a project must be completed by day 12. The last activity in the project consists of three consecutive day shifts\* and would need to finish on day 12 (5 PM). Cascading backwards, we note that the late start date for this activity would be day 10 (8 AM).

\*\*Eight-hour work periods from 8 AM to noon and 1 PM to 5 PM each day (one hour off for lunch). This is a standard default work day in Microsoft Project®.

# Backward Pass (Second Pass)

- A backward pass is sometimes called a *second pass*.

# Backward Pass Placement in Node

- In a precedence diagram node (activity), the late finish often appears in the lower right-hand part of the node; the late start often appears in the lower left-hand part of that node.\*

	Duration	
Activity Name		
<b>Late Start</b>		<b>Late Finish</b>

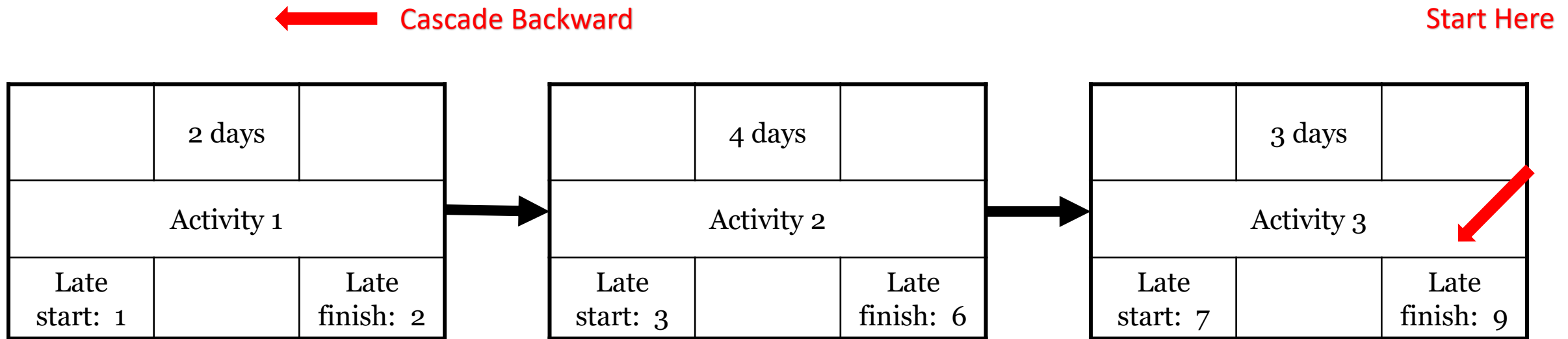
\*There is no standard for where information is placed on a node or what information appears on a node.

**Figure 6-38: Backward Pass Placement in Node**



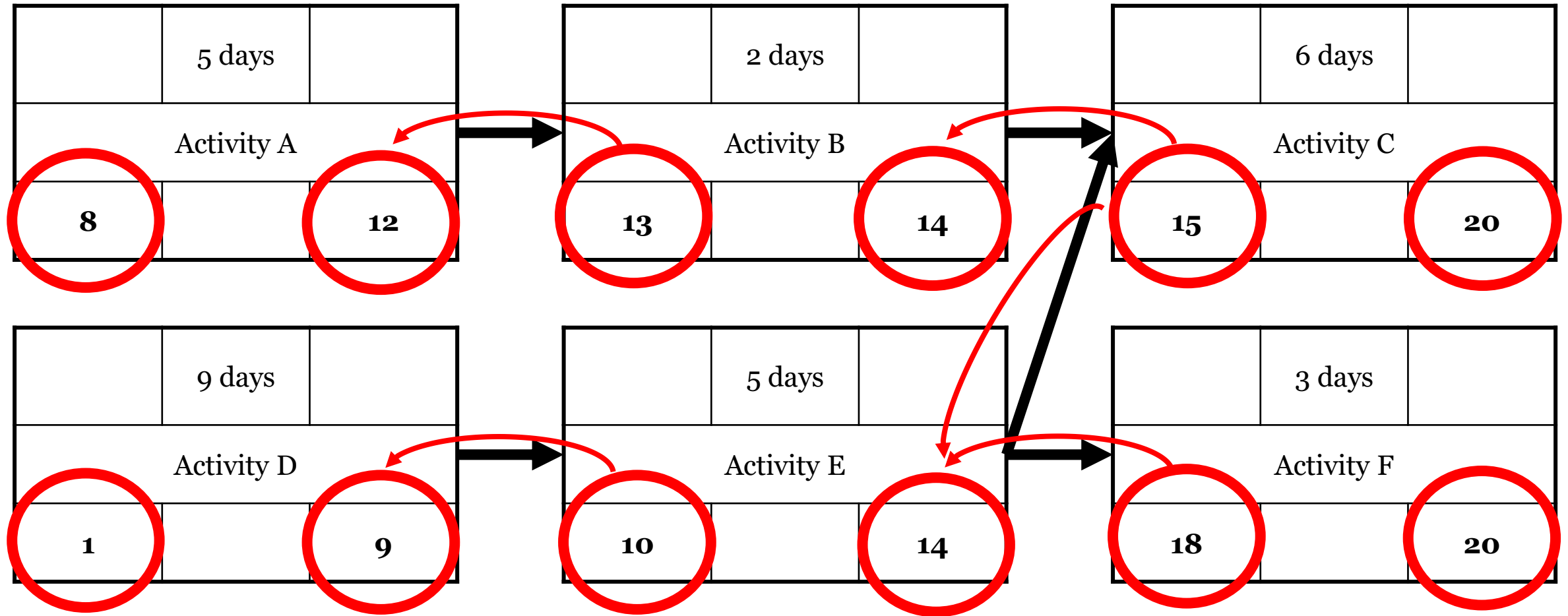
# Backward Pass Cascading Example One

- Note the late finishes and late starts for each node.



**Figure 6-39: Backward Pass Cascading Example**

# Backward Pass Cascading Example Two



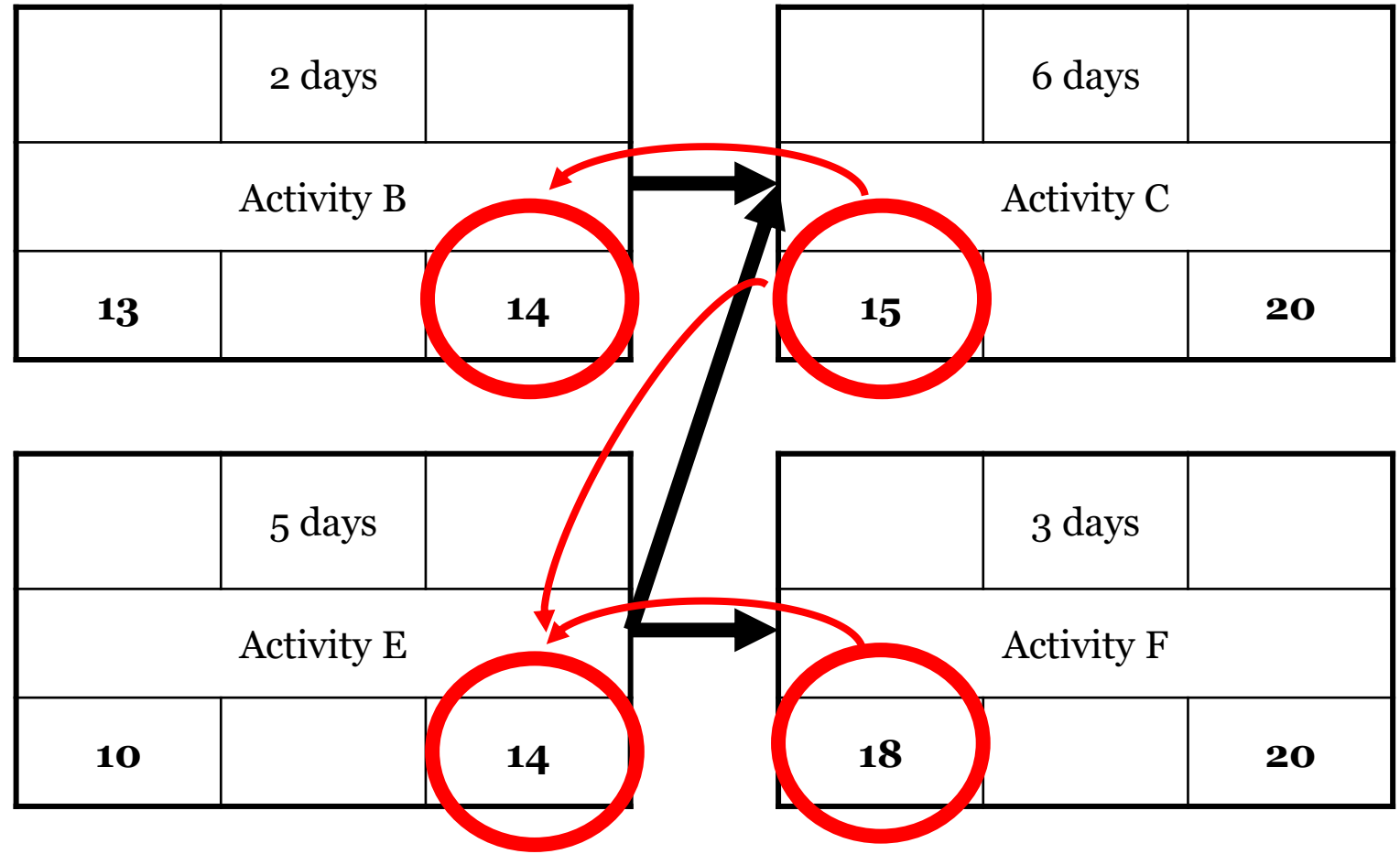
**Figure 6-40: Backward Pass Cascading Example**

# Backward Pass Dependencies

Note that the late finish for Activity E is day 14. This is because Activity E has two successors, Activity C and Activity F.

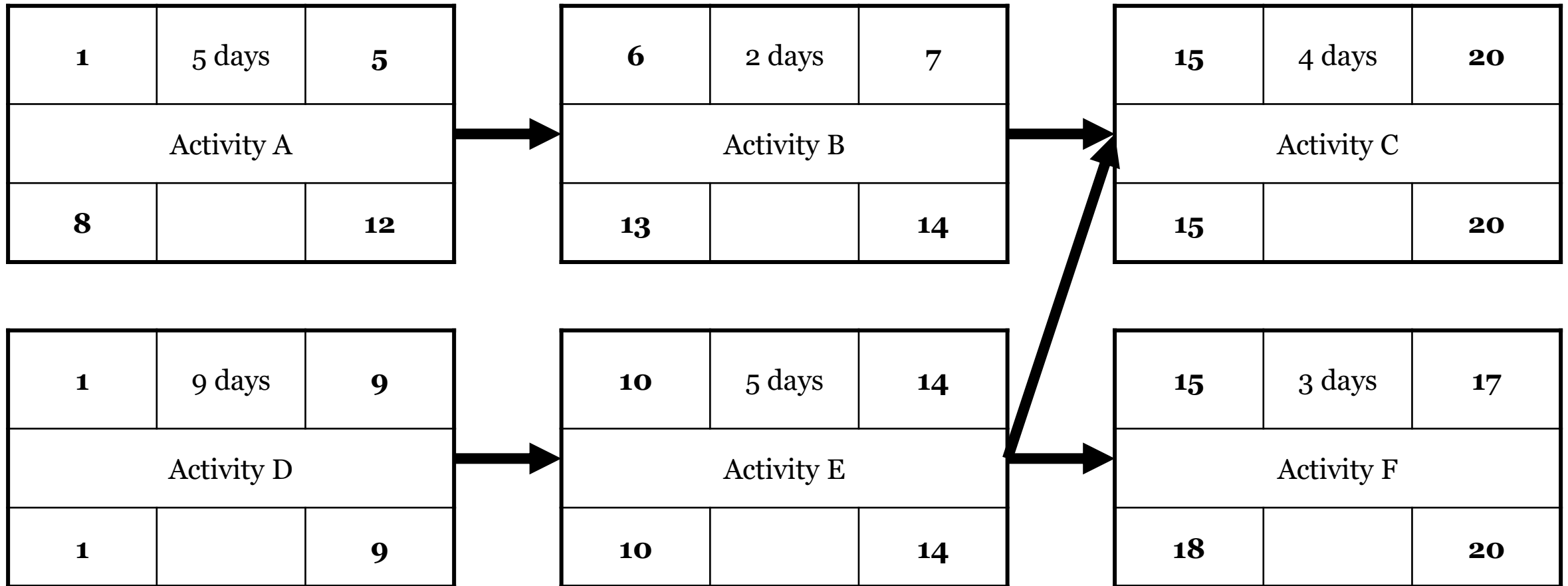
In the event that an activity has more than one successor, the predecessor activity late finish date (Activity E in this example) must be earlier than the earliest late start date of any successor activity (Activities C and F in this example).

Activity C's late start date is day 15, and Activity F's late start date is day 18. Hence, Activity E's late finish date must be day 14.



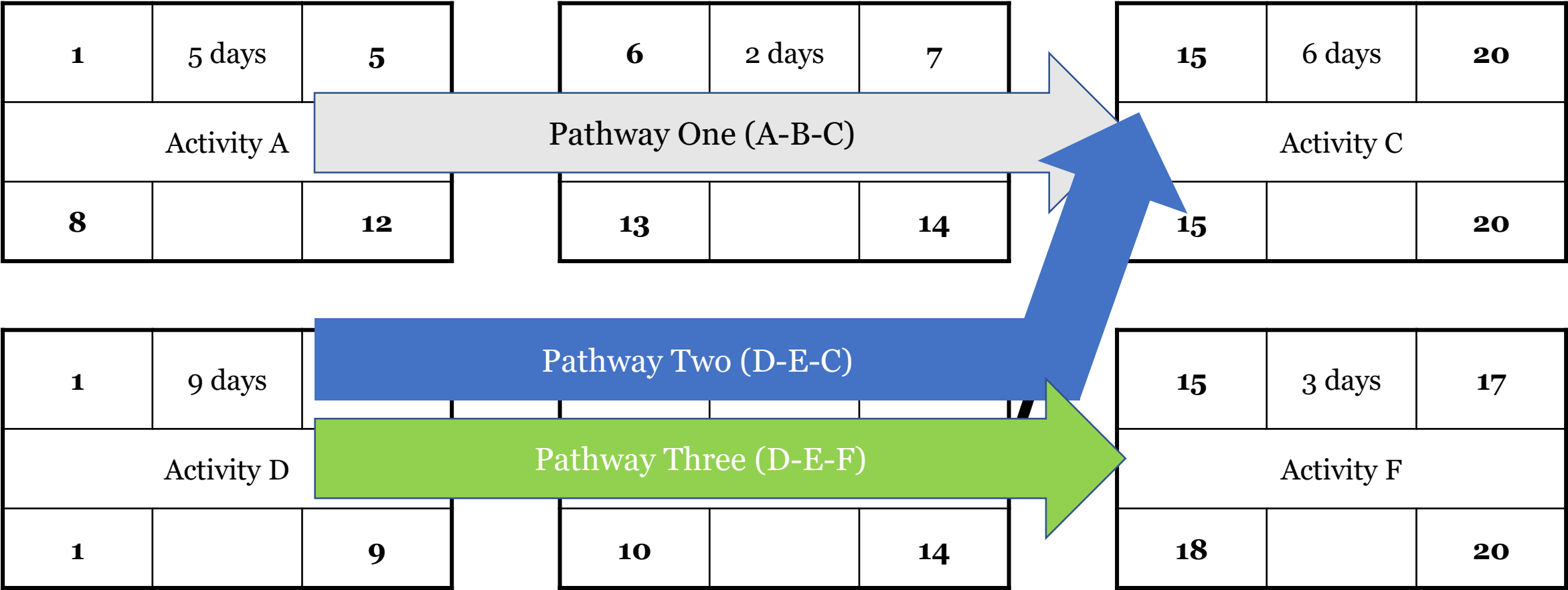
**Figure 6-41: Backward Pass Dependencies**

# Schedule Network Diagram Example



**Figure 6-42: Schedule Network Diagram Example**

# Three Pathways in this Example

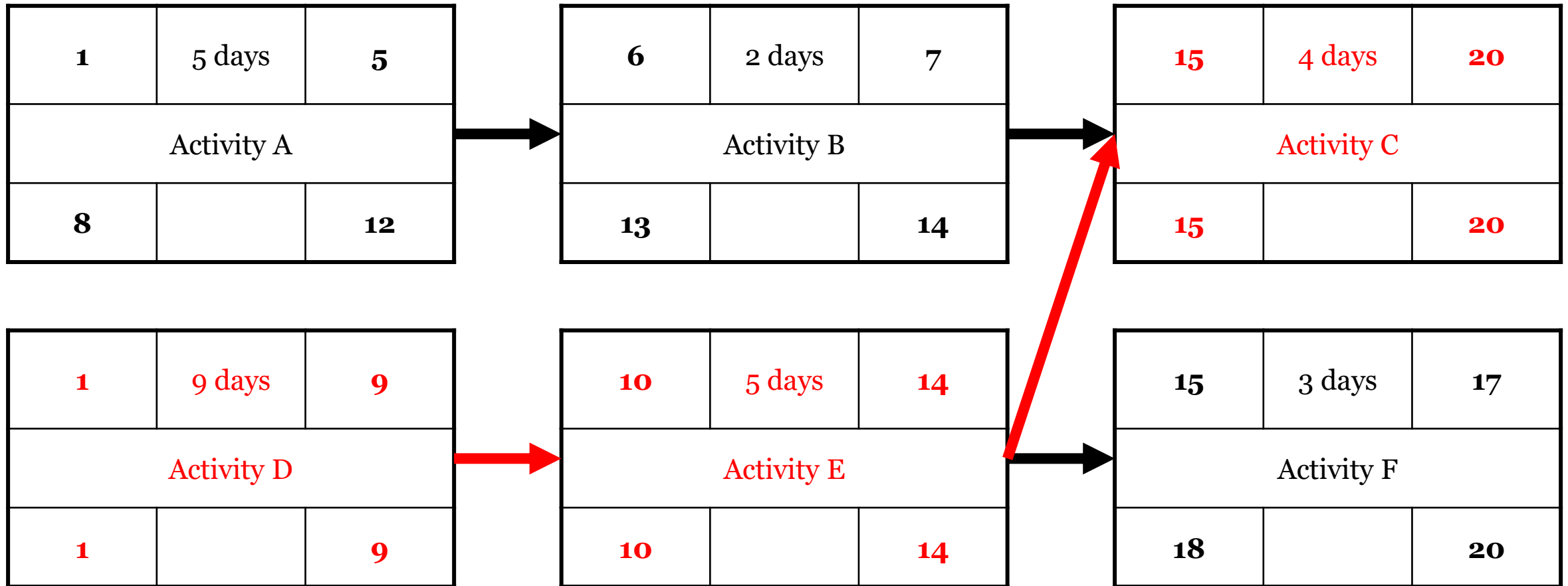


**Figure 6-43: Three Pathways in this Example**

# Critical Pathway

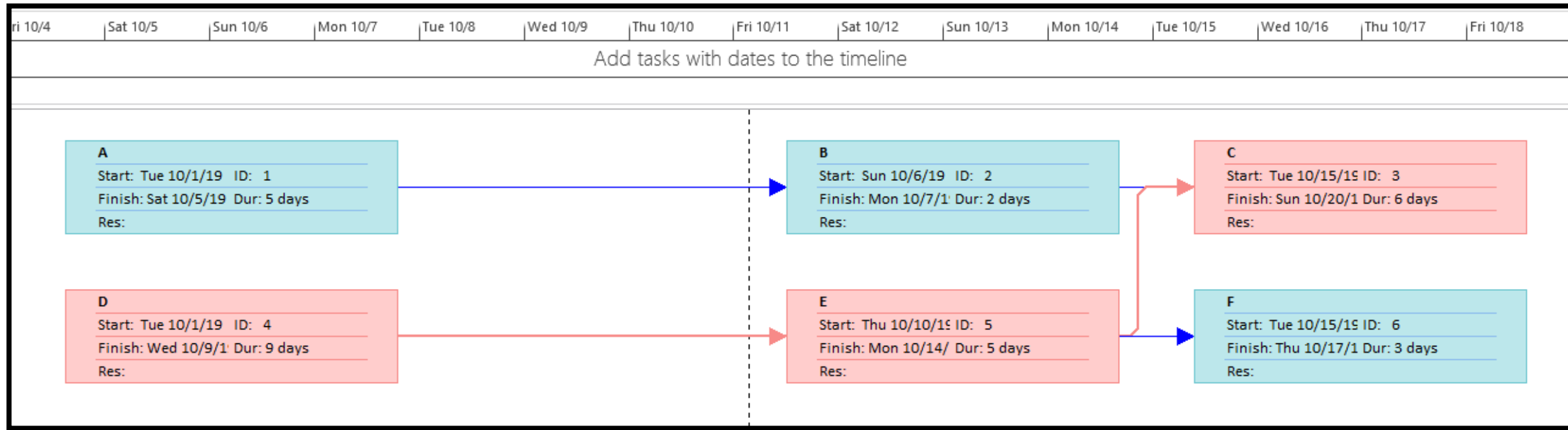
- One way to calculate a critical pathway is to sum the duration of each activity in each pathway.
- In the previous example, the pathway durations are as follows:
  - Pathway A-B-C = 5 days + 2 days + 6 days = 13 days total.
  - Pathway D-E-C = 9 days + 5 days + 6 days = 20 days total.
  - Pathway D-E-F = 9 days + 5 days + 3 days = 17 days total.
- Hence, the critical pathway is D-E-C (20 days).

# Critical Pathway



**Figure 6-44: Critical Pathway**

# Microsoft Project® Network Diagram Example



Red nodes and lines represent a critical pathway.

Figure 6-45: Microsoft Project® Network Diagram Example



# Float

# Float Overview

- There are two types of float that can be calculated using the results of the forward and backward passes:
  - Total
  - Free
- Float can help us determine how it might be possible to shorten a project schedule and redistribute activity resources.
- Float is sometimes called slack. These terms are used interchangeably.

# Total Float

- Total float tells us how long we can delay an individual activity without delaying the end of the project (this may be zero).
- Total float is the most commonly calculated float.
- Total float is calculated for each activity. In each activity, subtract its early start from its early finish or subtract its late start from its late finish.

# Total Float Placement in Node

- In a precedence diagram node (activity), the total float often appears in the bottom center of that node.\*

Activity Name		
	<b>Total Float</b>	

\*There is no standard for where information is placed on a node or what information appears on a node.

**Figure 6-46: Total Float Placement in Node**

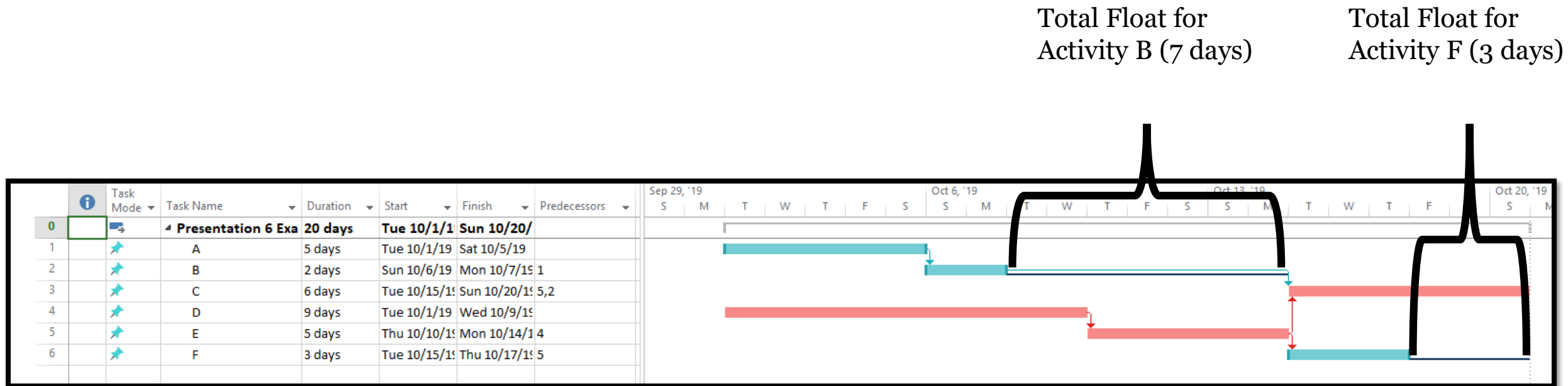
# Calculating Total Float Example

Start Formula	Finish Formula
Total Float = Late Start – Early Start	Total Float = Late Finish – Early Finish

1	5	5
	A	
8	7	12

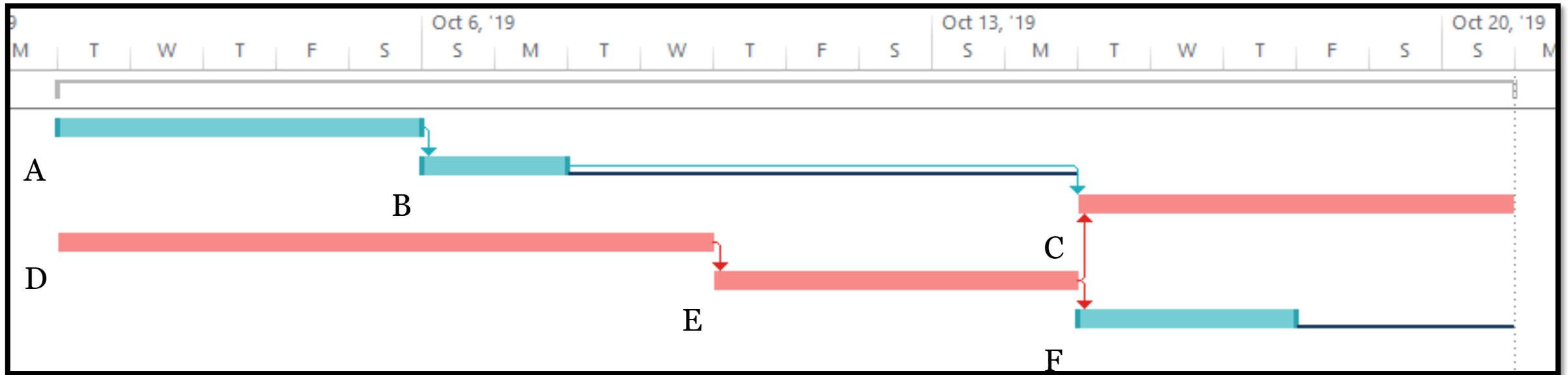
$$8 - 1 = 7 \text{ or } 12 - 5 = 7$$

**Figure 6-47: Calculating Total Float Example**



Red nodes and lines represent a critical pathway.

**Figure 6-48: Microsoft Project® Gantt Chart Example**



Red nodes and lines represent a critical pathway.

**Figure 6-49: Microsoft Project® Gantt Chart Slack Example Closeup**

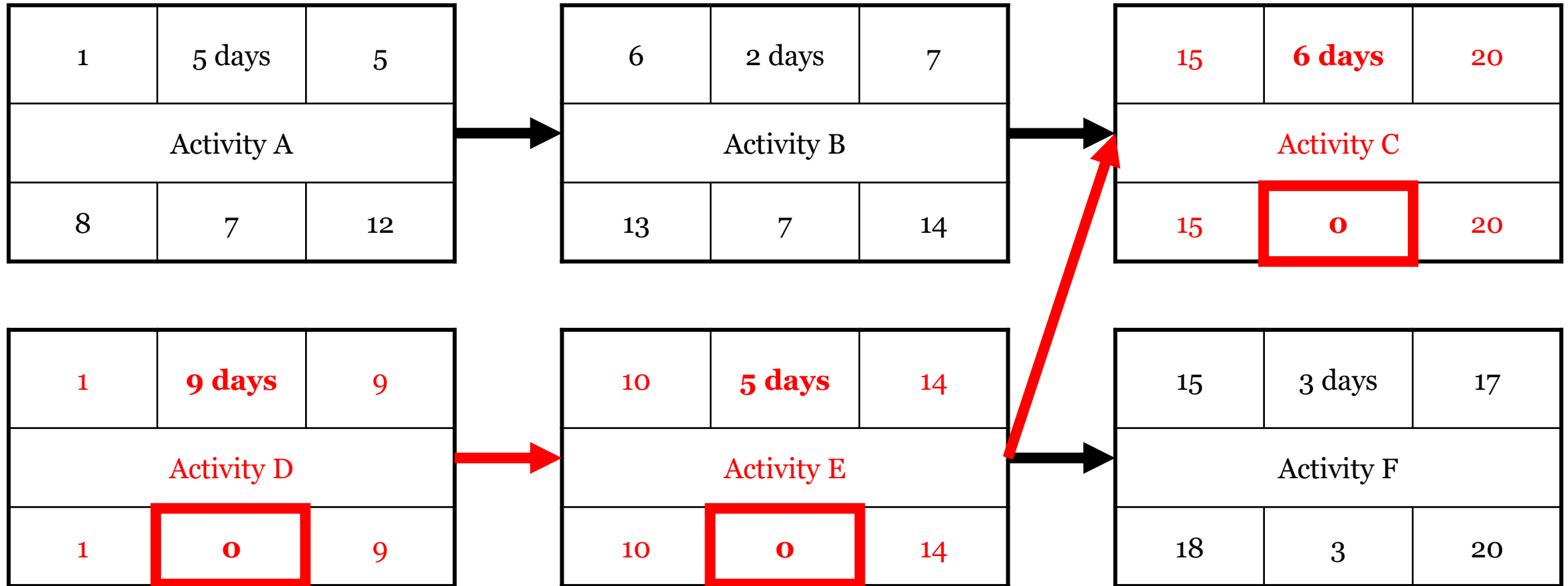
# Critical Pathway Example Notes

- In our example, in order to compress the schedule, Activities D, E, or C must be reduced in duration (e.g., adding more resources).
- For purposes of calculating a critical pathway, only one estimate per activity is used (as opposed to a range for each activity). Examples of time estimates include realistic, most likely, and expected value.
- Recall that each activity on a critical pathway will have zero total float (slack).



# Critical Pathway and Float Example\*

\*Note zero total float for every activity on a critical pathway.



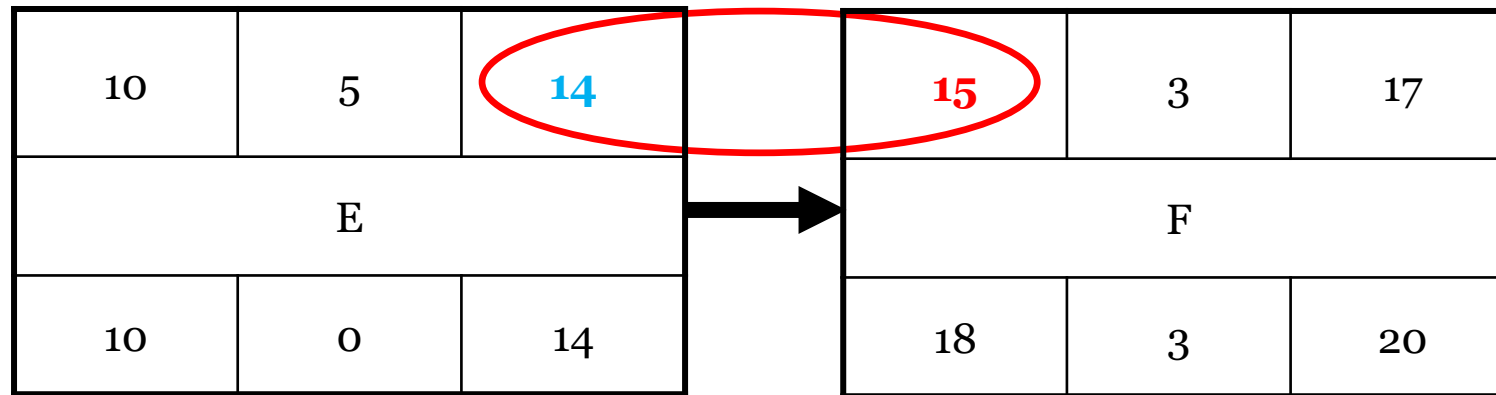
**Figure 6-50: Critical Pathway and Float Example**

# Free Float

- Another kind of float is free float.
- Free float tells us how long you can delay a predecessor activity without delaying a successor activity.
- Free float always involves at least two activities.
- Also note that free float does not normally appear on a node.
- Free float is calculated by adding one to the early finish of a predecessor activity. We then subtract that result from the early start date of its successor activity.

# Calculating Free Float Example

Formula
Add 1 (one) to the <b>early finish</b> of the predecessor activity. Subtract that result from the <b>early start</b> of the successor activity.



$$15 - (14 + 1) = 0$$

**Figure 6-51: Calculating Free Float Example**

# Importance of Free Float

- If free float exists in a schedule, it might be possible to extract resources from those activities with free float without extending the project finish date.

# Project Float

- Project float tells us how long a project can be delayed without delaying an externally enforced project deadline (i.e., completion date) from the customer, sponsor, etc.

# Schedule Baseline

- The combination of all the outputs of these methodologies leads to the creation of the schedule baseline. It is primarily derived from schedule network analysis.
- It is the official schedule authorized by the project management decision making authority, and it reflects planned start and finish dates for all activities as well as milestone dates.

# Resource Optimization

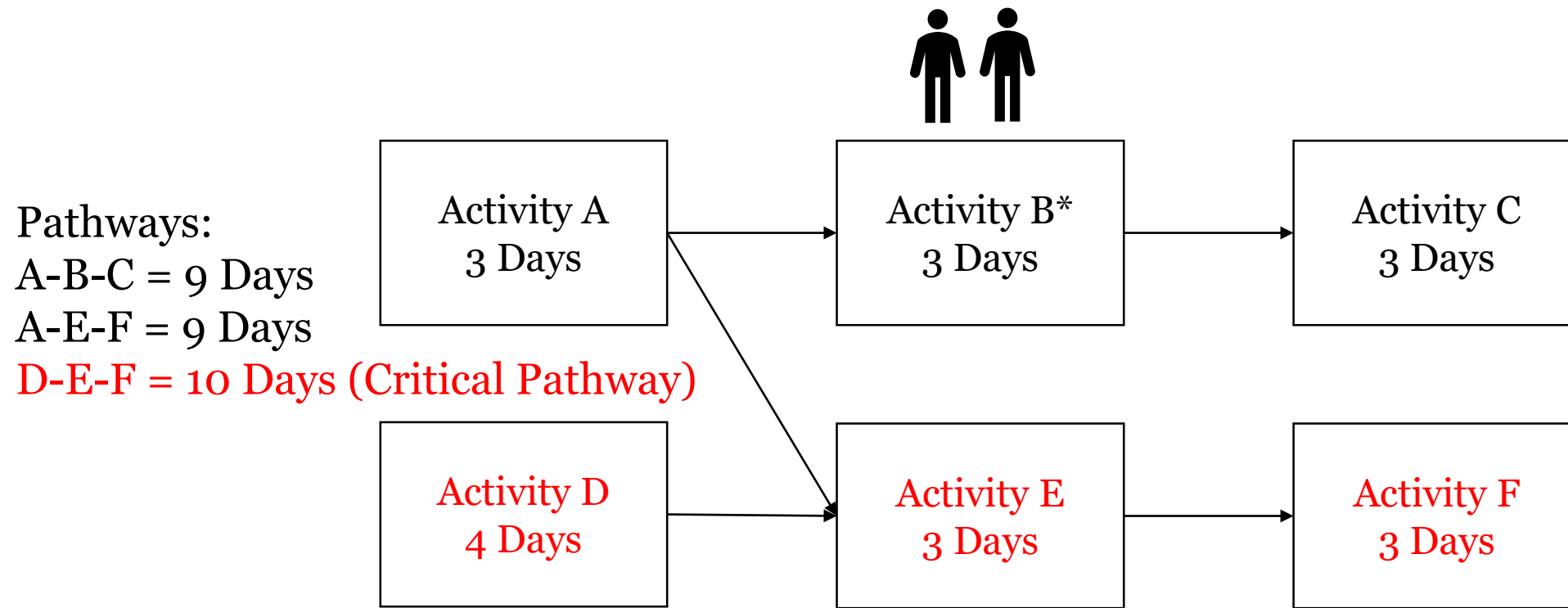
# Resource Optimization

- Resource optimization involves calculating the most efficient and effective use of resources within a project given the resources available and schedule constraints.
- Two popular methods are resource leveling and resource smoothing.



# Resource Leveling

- Resource leveling involves the process of altering start and finish dates for individual activities in relation to the availability of resources.
- Depending on the availability of resources, leveling may change a critical pathway.
- If a critical pathway is altered with resource leveling, the project finish date will be extended.



\*Before leveling, Activity B has two technicians as a resource. With two technicians, it takes 3 days to complete Activity B. Currently, it takes 10 days to complete the project.

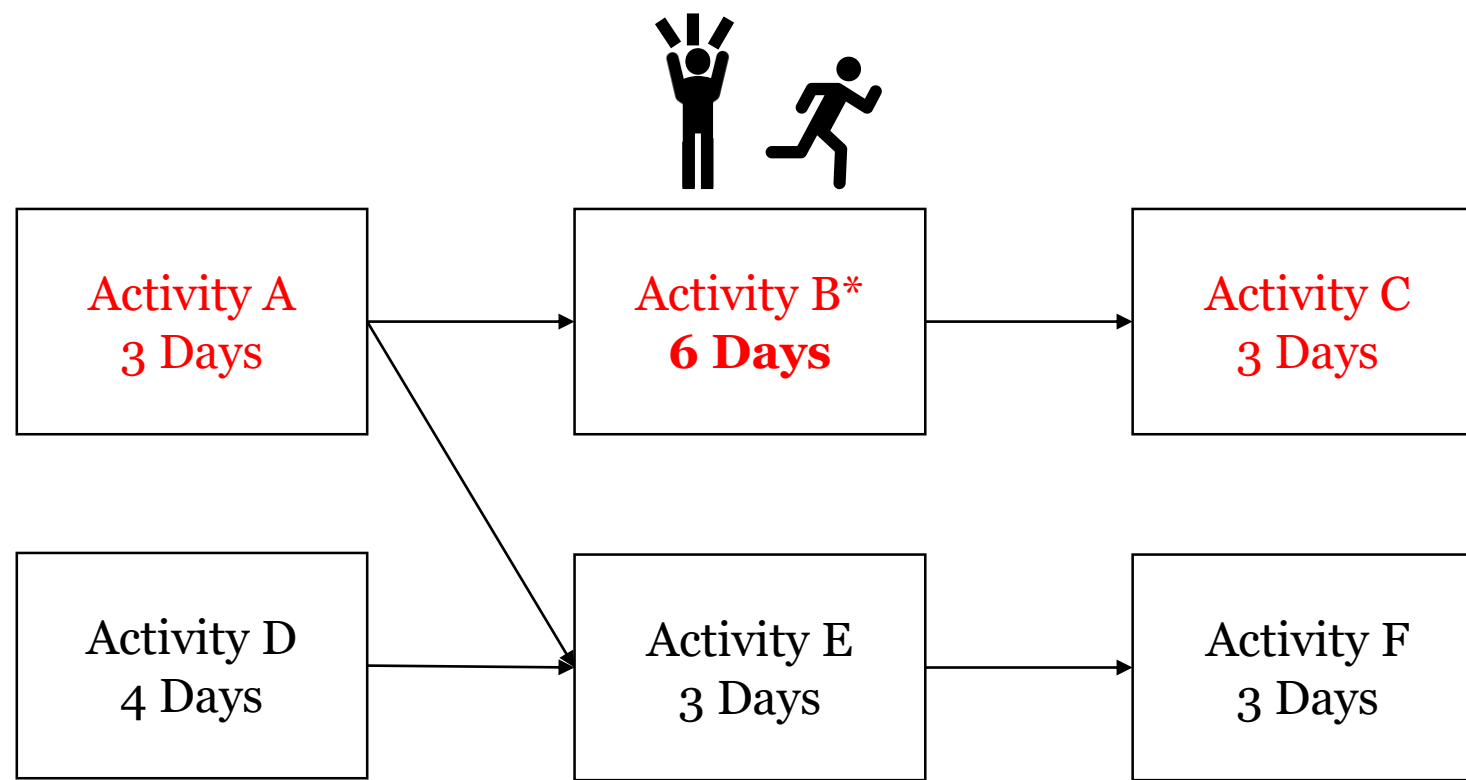
**Figure 6-52: Resource Leveling Example Part 1**

Pathways:

**A-B-C = 12 Days**

A-E-F = 9 Days

D-E-F = 10 Days



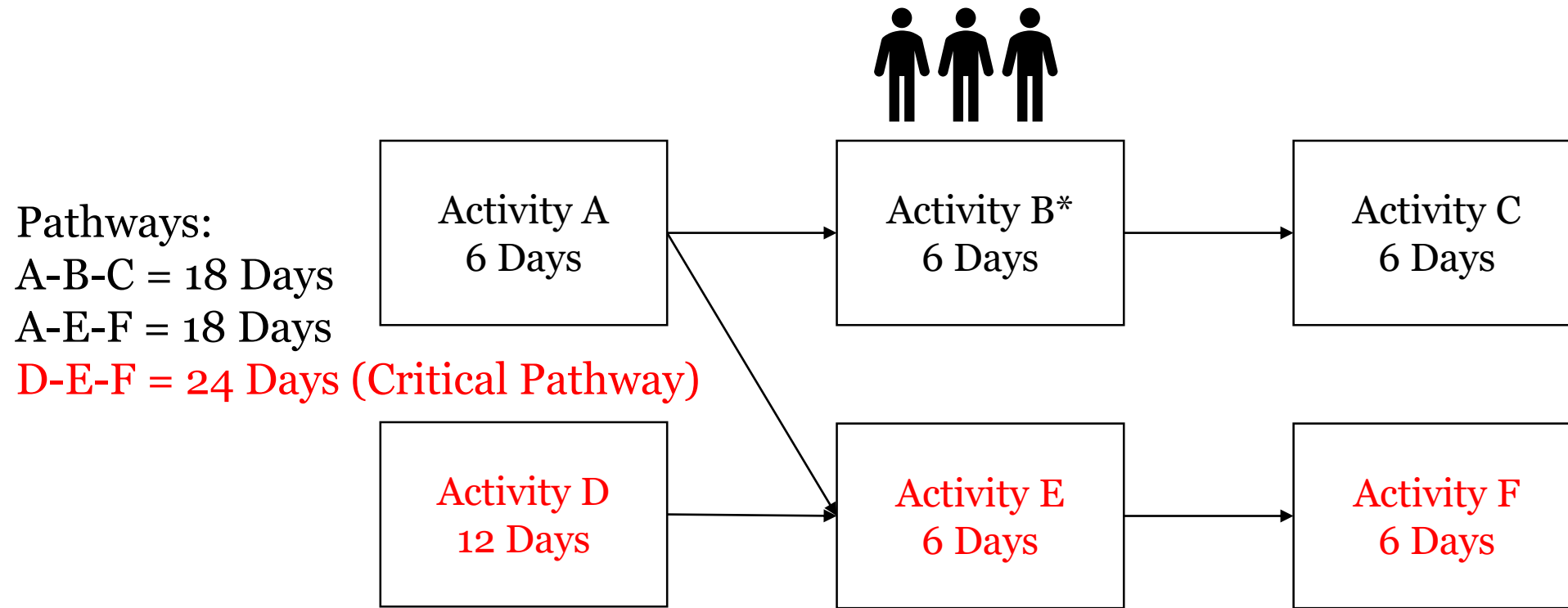
\*A change in the organization requires that one of the two technicians assigned to Activity B be moved to another project. With the remaining technician in Activity B, it now takes 6 days (instead of 3 days) to complete Activity B.

Note: The critical pathway changes (from D-E-F to **A-B-C**), and the overall project duration changes to 12 days.

**Figure 6-53: Resource Leveling Example Part 2**

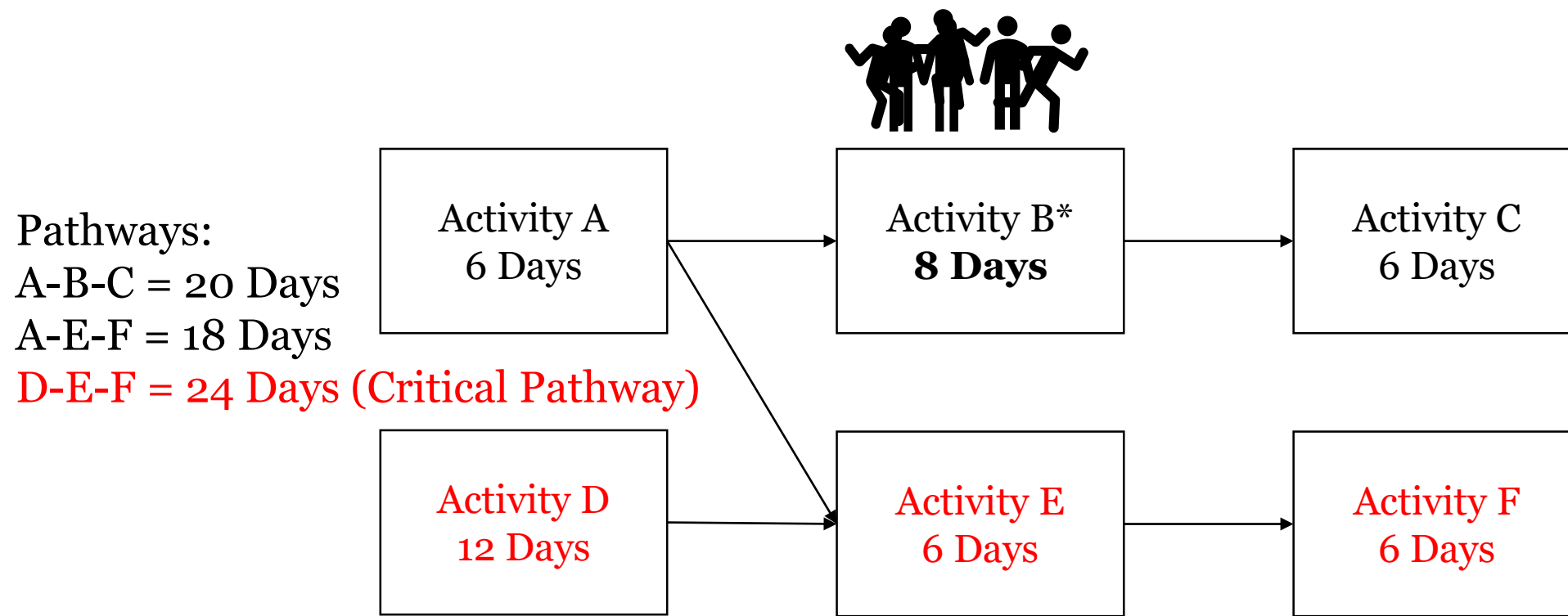
# Resource Smoothing

- Resource smoothing involves a technique that adjusts resource expenditures based on a pre-defined resource authorization limit.
- In this instance, the critical pathway is not changed.
- The project due date is not changed (delayed).



\*Before smoothing, Activity B had three technicians as a resource. With three technicians, it takes 6 days to complete Activity B. Currently, it takes 24 days to complete the project.

**Figure 6-54: Resource Smoothing Example Part 1**

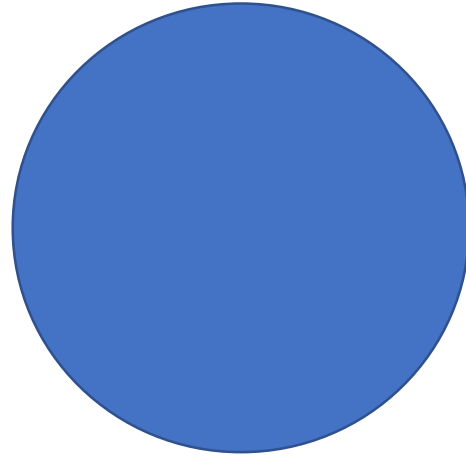


\*As it turns out, one of the technicians assigned to Activity B will earn overtime pay if that technician works on Activity B. Overtime pay is not authorized on this project, and this technician is removed from the project. With the remaining two technicians in Activity B, it now takes 8 days to complete Activity B. Note: Neither the critical pathway nor the overall project duration changes.

**Figure 6-54: Resource Smoothing Example Part 2**

# Reserve Analysis

- Once we have determined how long an activity takes to accomplish, we need to determine the amount of time reserves (i.e., buffers) needed based on the amount of risk associated with the project.
- These buffers will help prevent bottlenecks and work stoppages.
- We give a special emphasis to critical pathways and near-critical pathways.



# Schedule Compression



# Schedule Compression

- After we have created a schedule, there are methods we can apply in an attempt to compress it.
- Two basic types are crashing and fast tracking.
- In conjunction with these compression techniques, we can also conduct a what-if analysis (e.g., Monte Carlo) to assess schedule risk.

# Crashing

- One approach to reducing a schedule is to add additional resources to certain activities to reduce the duration of those activities.
- This is a studied approach; do not forget that adding resources to an activity is not a linear process. At some point, we achieve diminishing returns when we add resources.

# Fast Tracking

- Fast tracking involves conducting one or more phases or activities in parallel to speed up the final delivery date of the project.
- Obviously, for this strategy to work, phases conducted in parallel cannot have concurrent dependencies.

# Monte Carlo and Schedule Risk Management

# Monte Carlo Analysis (Simulation)

- Monte Carlo is a type of what-if analysis created in the 1940s by physicists who were trying to calculate the probabilities of what would happen in a highly-enriched uranium chain reaction.
- What-if scenario analysis is a manipulation of one or more of the model variables to determine how that manipulation would influence a schedule.
- Monte Carlo uses modeling (e.g., project management software) to represent the use and availability of resources to project schedule duration.

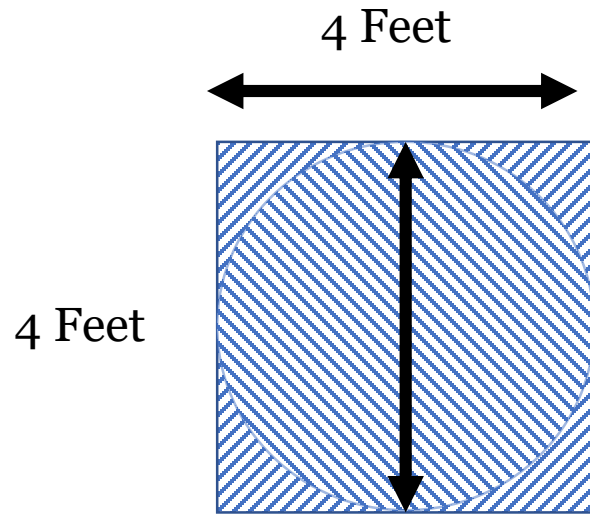
# Monte Carlo and Risk Management

- Monte Carlo is a type of simulation that produces estimates for each activity.
- These activity estimates are then used to calculate an overall estimate of project duration.
- Monte Carlo provides a sensitivity analysis of activity variation. In this way, the projected impact of an activity's change can be recorded and accounted for.

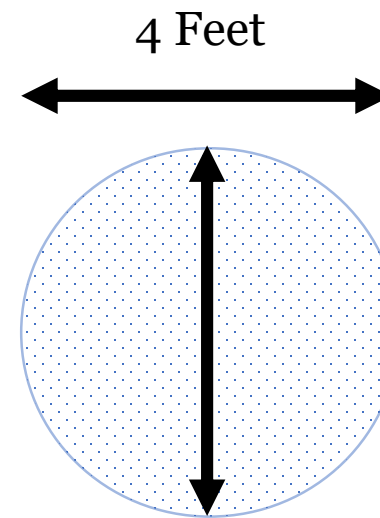
# Monte Carlo Example

- We want to calculate  $\pi$ .
- We know that a circle inscribed in a square will have the following ratio (with respect to area)  $\pi \div 4$ .
- We can use a Monte Carlo simulation to calculate  $\pi$ .

# Circle Inscribed in a Square Example



Square Area = 16 Square Feet



Circle Area = 12.56637 Square Feet

$$\begin{aligned} 12.56637 / 16 &= 0.785398125 \\ 0.785398125 \times 4 &= 3.1415925 \text{ (an approximation of } \pi) \\ \text{or} \\ 12.56637 / 4 &= 3.1415925 \text{ (an approximation of } \pi) \end{aligned}$$

**Figure 6-55: Circle Inscribed in a Square**



# Monte Carlo Example

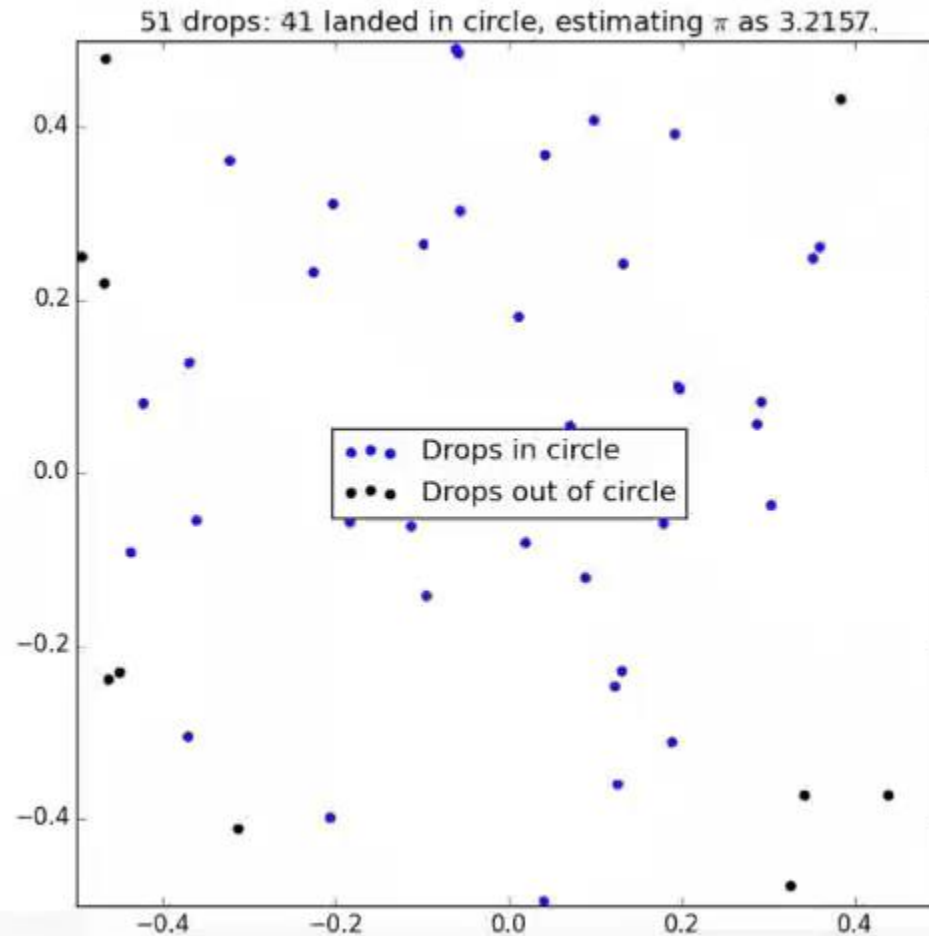
- A Monte Carlo can help us determine the area of a circle (relative to square in is circumscribed on). That is, what percentage of that square's area is circumscribed by the circle.
- Drop objects of a uniform size (e.g., grains of sand) in a random way over the square (these are the inputs).
- Count the number of objects inside the circle.
- Count the total number of objects dropped.
- The ratio created by the two areas will be approximately equal to  $\pi \div 4$  (multiply result by 4).

# Monte Carlo Example

- Drop 10,000 grains of sand onto the square.
- If the drops are truly random, about 7,854 of them should land inside the circle or 78.54%
- Multiplying this result by 4 yields 3.1416.

# Monte Carlo Example Notes

- The inputs (e.g., dropping of grains of sand) must be truly random.
- If the decimal representation is a non-repeating pattern, larger Monte Carlo samples yield more accurate results.
  - Example: 10,000 grains of sand  $\pi \approx 3.1416$
  - Example: 10,000,000 grains of sand  $\pi \approx 3.1415927$

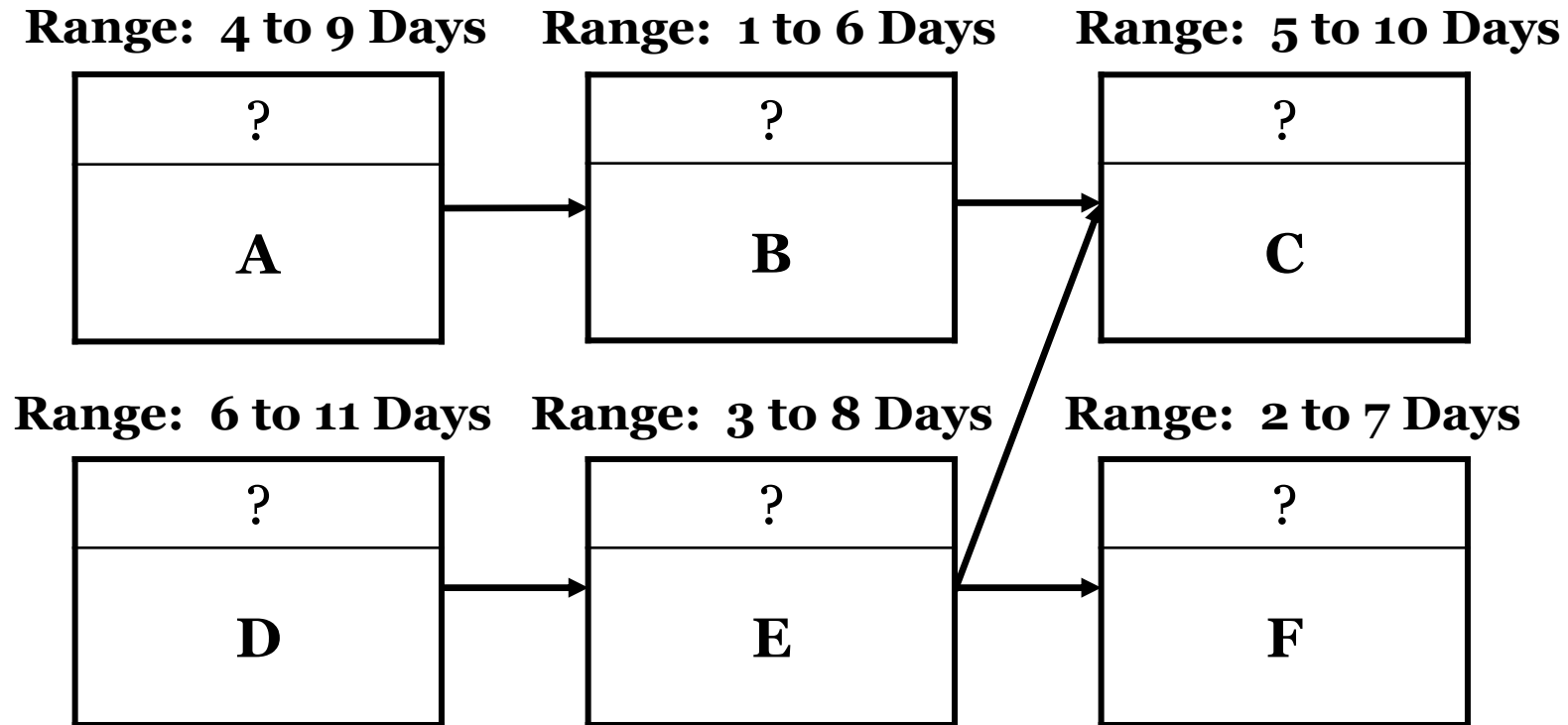


**Figure 6-58: Random Dropping of Uniform Objects in a Square**

# Modeling Schedule Risk Using Monte Carlo

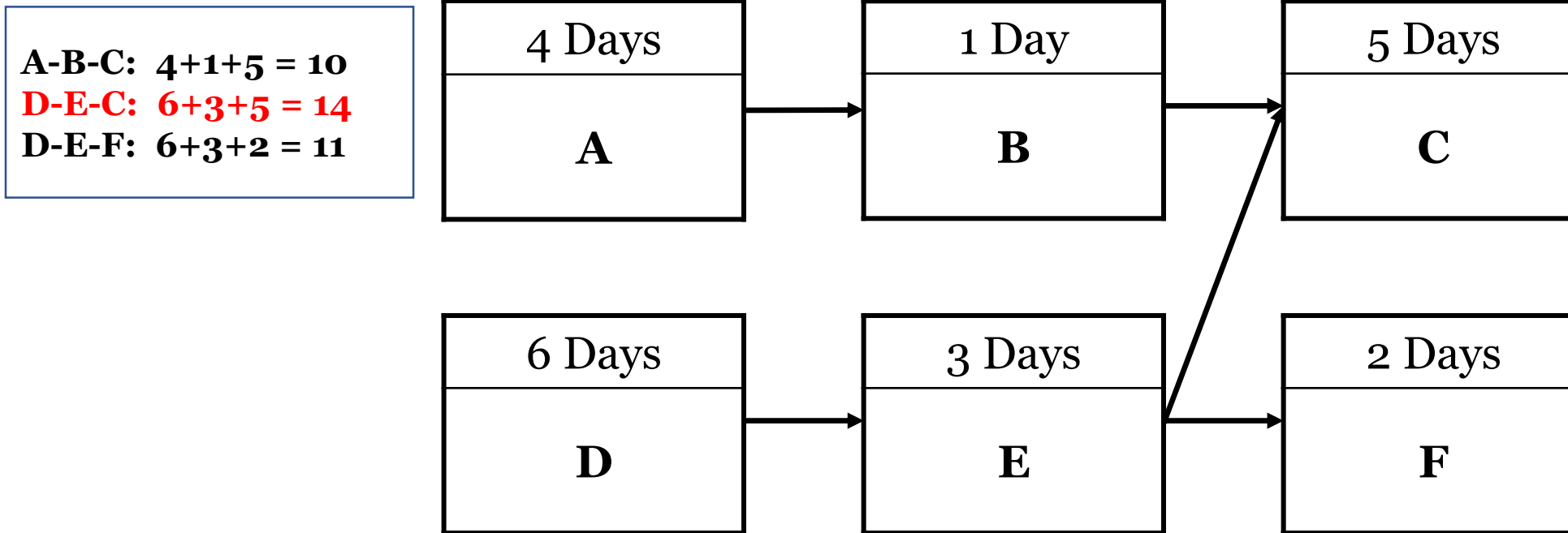
- In this example, there are six activities.

# Duration Possibilities for Each Activity



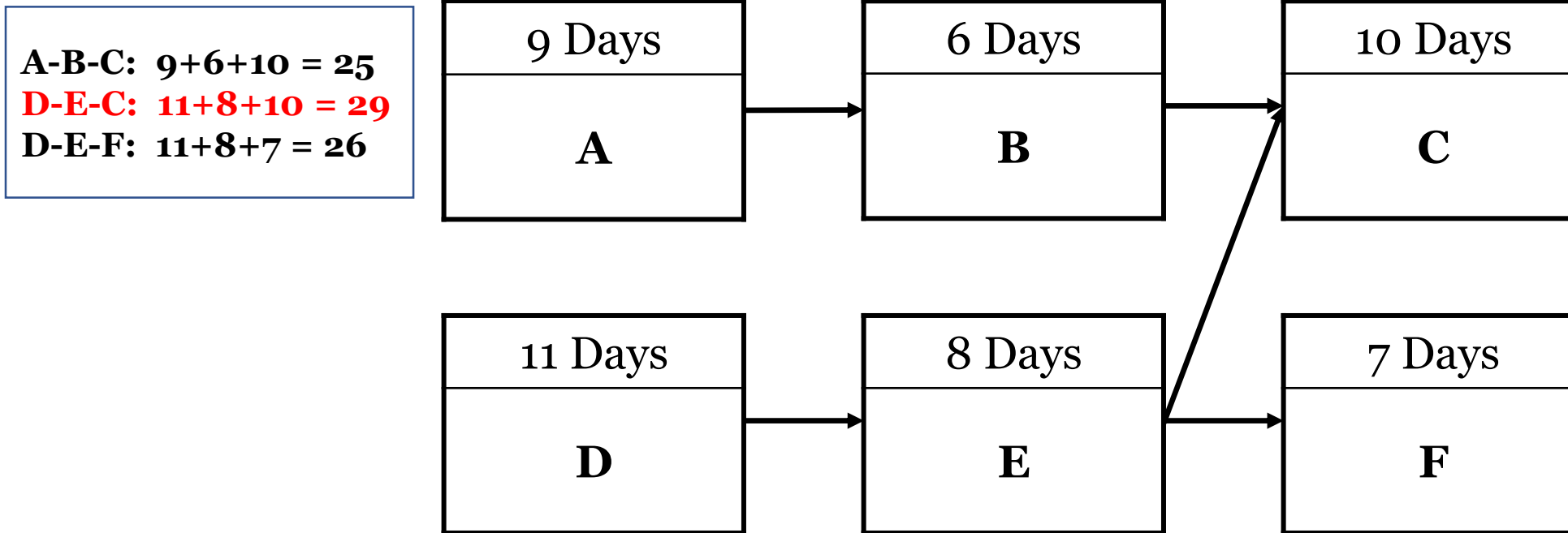
**Figure 6-59: Range of Duration Possibilities for Each Activity**

# Low Duration Possibilities for Each Activity



**Figure 6-60: Low Duration Possibilities for Each Activity**

# High Duration Possibilities for Each Activity



**Figure 6-61: High Duration Possibilities for Each Activity**

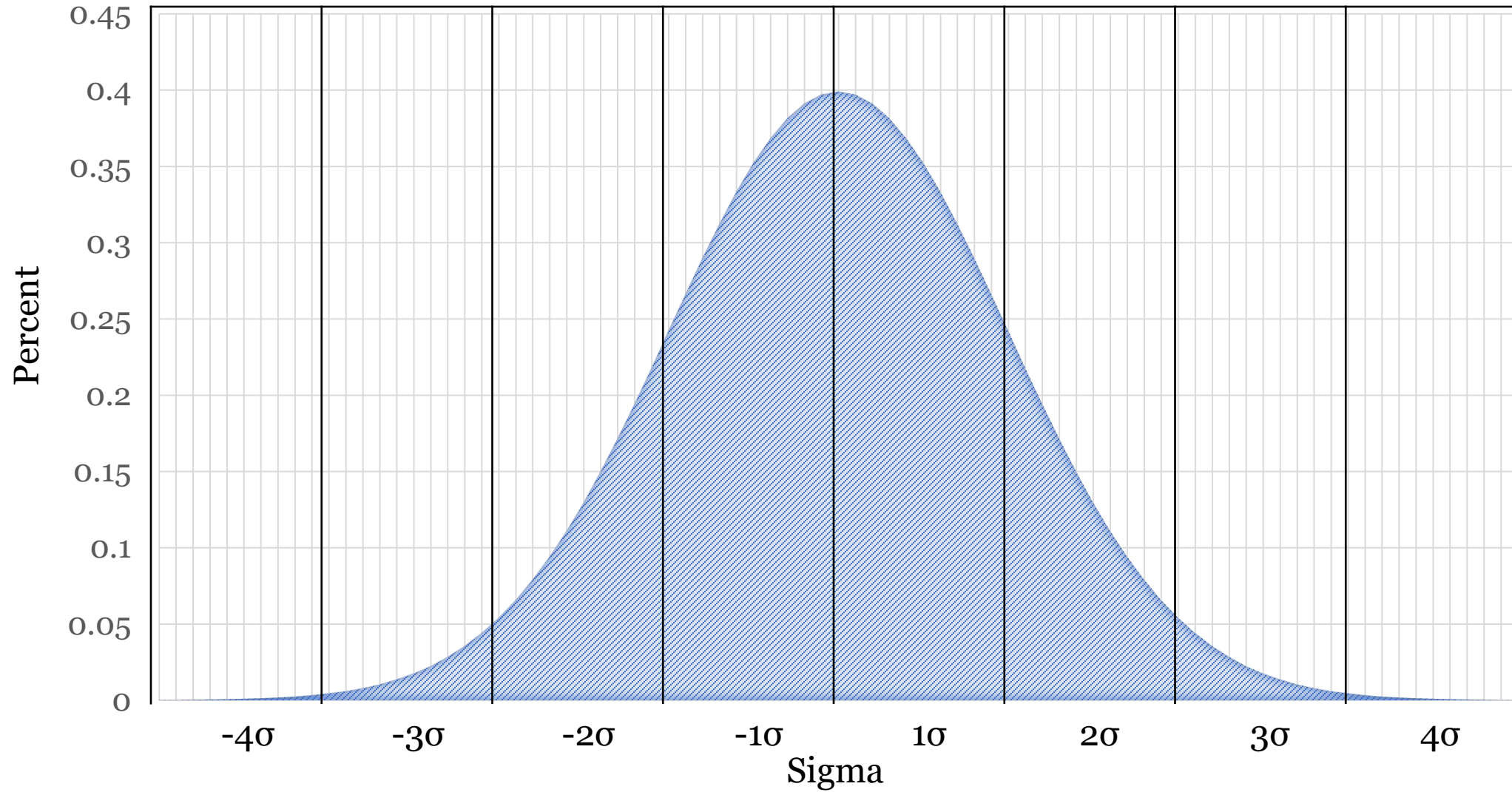


# Activity Duration Assumptions

- Each activity has a range of six days. For example, Activity A can be from four to nine days long. Activity D can be from five to eleven days long and so on.
- We assume that the duration distribution for each activity follows a normal distribution. In other words, it is more likely that Activity's A duration will be six days long (34.135%) than four days long (2.14%). See the probability assumptions for each activity on the next slide.

		Activity Durations					
Sigma	Probability	A	B	C	D	E	F
-4 $\sigma$	.135%						
-3 $\sigma$	2.14%	4	1	5	6	3	2
-2 $\sigma$	13.59%	5	2	6	7	4	3
-1 $\sigma$	34.135%	6	3	7	8	5	4
1 $\sigma$	34.135%	7	4	8	9	6	5
2 $\sigma$	13.59%	8	5	9	10	7	6
3 $\sigma$	2.14%	9	6	10	11	8	7
4 $\sigma$	.135%						
	100%						

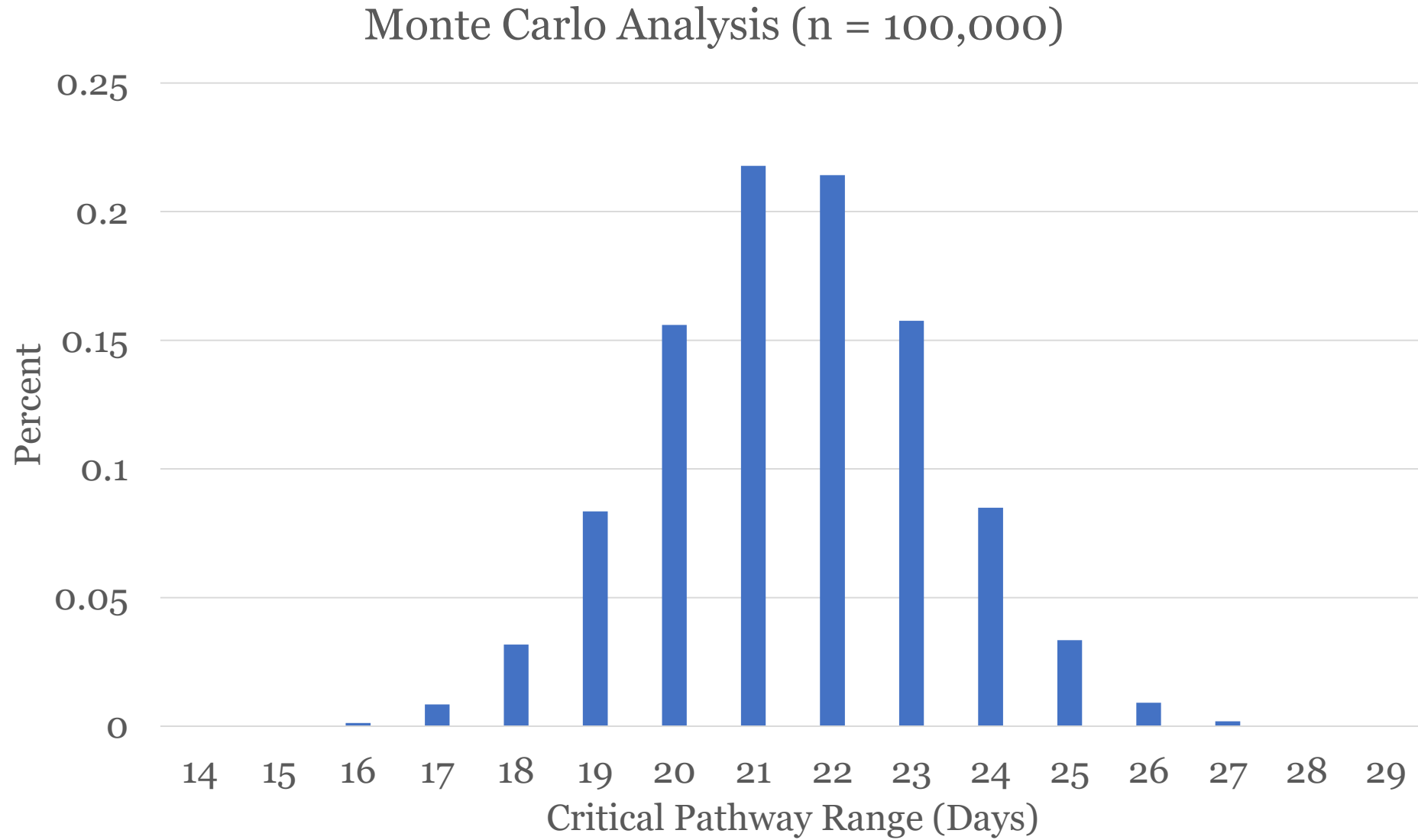
**Table 1: Activity Duration Assumptions Table**



**Figure 6: Normal Distribution and Sigma**

# Monte Carlo Analysis Assumptions

- For each model, each activity's duration is randomly generated based on a normal distribution. For example, on any given model, Activity A has a 34.1% chance of a 6 day duration.
- The data were modeled randomly 100,000 times, and the data were then summed. The cumulation approximates a bell curve (expected).



**Figure 6-64: Monte Carlo Critical Pathway Range Days**

# Monte Carlo Simulation and Program Evaluation and Review Technique (PERT) Comparison

- We now compare our Monte Carlo simulation to a PERT analysis using the same data to see how the results compare.

# PERT Analysis

- We will conduct a PERT analysis on the critical pathway (D-E-C) using the low, high, and most likely estimates.
- We will assume the range of possible dates for each activity follows a normal distribution. Hence, it follows that an approximation of the most likely estimate for each activity is the mean of the range of those possibilities. For example, Activity A can be 4, 5, 6, 7, 8, or 9 days long. We will assume the mean of the sum of these possibilities (6.5 days) is the most likely estimate.

# PERT Analysis on the Critical Pathway

Model Information						
Activity	Low (o)	High (p)	Most Likely (m)	EV $(o + p + (4 \times m)) / 6$	SD $(p - o) / 6$	SD <sup>2</sup>
D	6	11	6.5	8.5	0.8333	0.6944
E	3	8	5.5	5.5	0.8333	.0.6944
C	5	10	7.5	7.5	0.8333	0.6944
Sums:				<b>21.5</b>		<b>2.0833</b>
Square Root of Sum:						<b>1.4434</b>

**Table 6-3. PERT Analysis on the Critical Pathway**

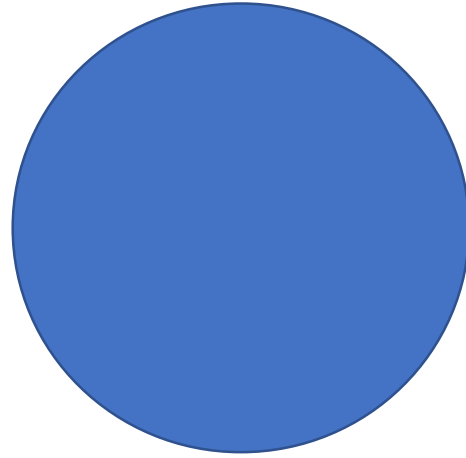


# PERT Analysis Results

- The expected value for the critical pathway is 21.5 days. One standard deviation for the critical pathway is 1.4434.
- Probability the project will be complete in these ranges (standard deviations):
  - 68.26% chance it will be complete between 20.06 to 22.94 days.
  - 95.44 % chance it will be complete between 18.61 to 24.39 days.
  - 99.73% chance it will be complete between 17.17 to 25.83 days.

# Comparison Analysis

- The PERT and Monte Carlo analyses yield similar results.
- Both PERT and Monte Carlo suggest that there is around a 95 to 96 percent probability that the project will finish at 24 days or less.
- Both PERT and Monte Carlo suggest that there is around a 99.73 percent probability that the project will finish at 26 days or less.



# A Summary of Time Management

# Managing the Schedule

- As the project schedule is executed, we will want to provide key stakeholders with information about schedule performance.
- This information will be used to help decision-makers and project team specialists meet schedule management goals.
- The timing and format of schedule performance information (sometimes referred to as work performance information) varies based on the needs of the stakeholders.

# Schedule Information Types

- The project management schedule outlines the proposed schedule activity dates and milestones. Some important types of scheduling information includes the following:
  - Bar charts
    - Gantt charts
  - Milestone charts
  - Resource histograms
  - Project schedule network diagrams (e.g., activity-on-node)
- Note: The project schedule must be finished before the release of the project management plan.

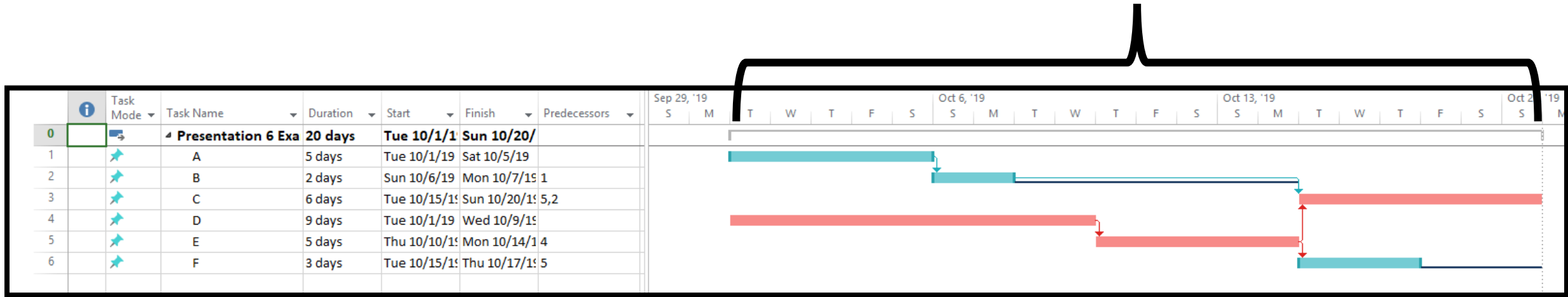
# Bar Chart

- Senior-level management stakeholders often receive information in a bar chart format.
- This chart is a graphical representation illustrating two bars for each activity. One of the bars displays the level of expected completion. The other bar shows the level of actual completion for the activity.
- A demarcation line often indicates work-in progress and is referred to as the as-of date or status date.

# Gantt Chart

- A popular type of bar chart is the Gantt chart.
- By comparing the relative levels of the bars, stakeholders can visually compare how far ahead or behind each activity is at any given point.
- To help manage the schedule, some work packages are grouped together to create an additional bar summarizing certain activities. This is sometimes referred to as a hammock activity.

Summary bar  
(hammock) for  
the entire project.



Red nodes and lines represent a critical pathway. Note: This project has not started. Hence, there is no progress to report or compare.

**Figure 6-65: Microsoft Project® Gantt Chart Hammock Example**



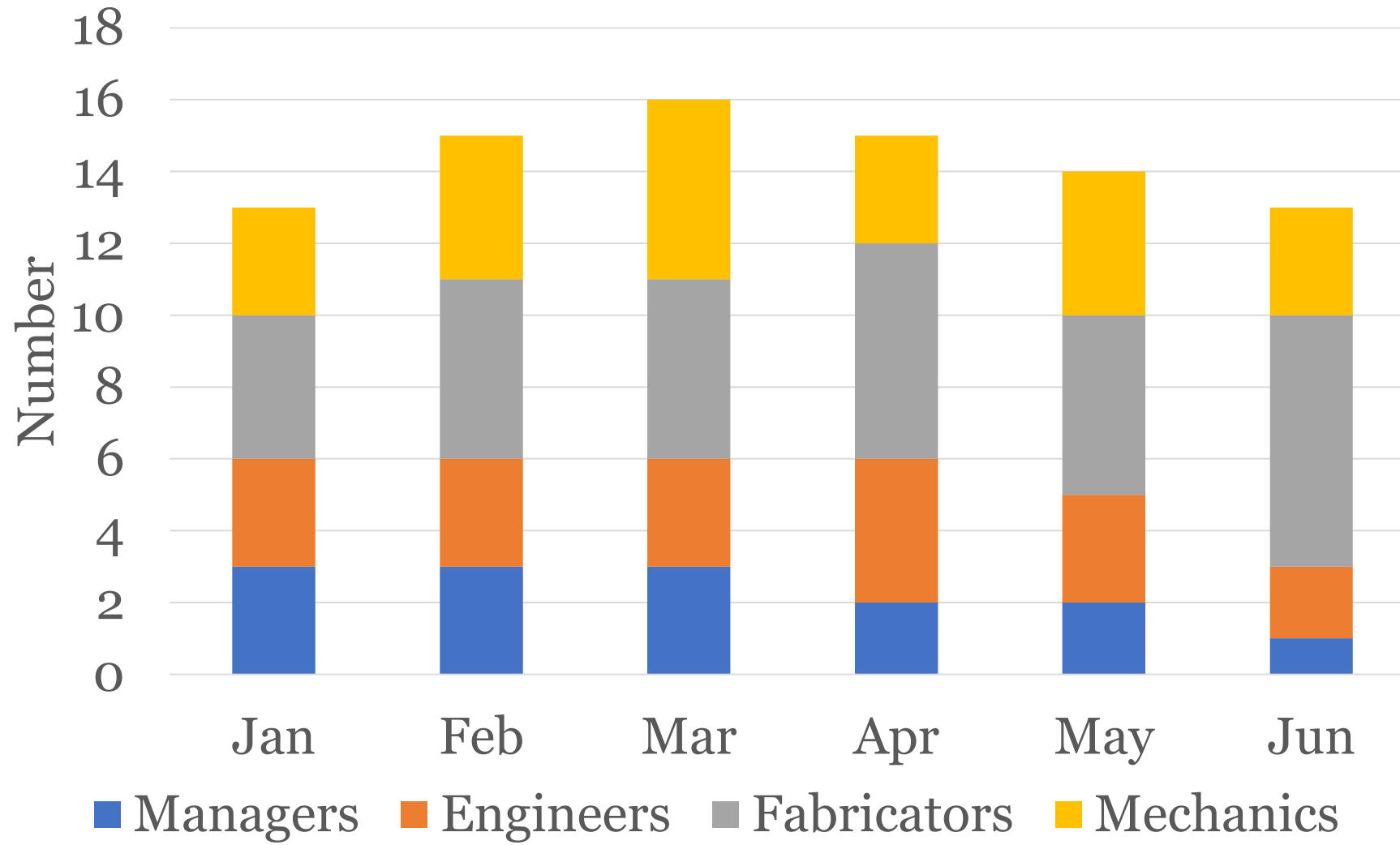
# Milestone Chart

- A milestone chart is similar to a bar chart, but it only illustrates the start and finish of key deliverables.

# Resource Histogram

- A resource histogram is another useful tool for illustrating project schedule information.
- It displays what resources will be used over time.
- It differs from a bar chart in that it contains discrete bars without a side-by-side comparison (e.g., two bars per activity).

# Resource Histogram



**Figure 6-66: Resource Histogram**

# Managing the Schedule: Change Management

- As the project is executed, we have to manage the schedule. Our first job is to assess where the schedule is at any given moment. In other words, what is our status.
- We will take actual schedule performance and compare that to the schedule baseline. If something has changed, we need to find assignable causes (what factors caused the changes).
- Once we have identified what factors are creating the changes, we need to manage these factors.
- If necessary, we apply and adjust buffers in pathways and chains.

# Agile Schedule Management

- At the end of each iteration, we will perform an analysis of what happened and how we might improve the project (known as a retrospective).
- The retrospective will include variance analysis. It can also lead to a change in priorities of work that remains in the backlog.
- Another important parameter we will assess is iteration velocity. Velocity tells us if deliverables are being finished at an acceptable rate.
- As with a traditional project management methodology, we also need to manage change.

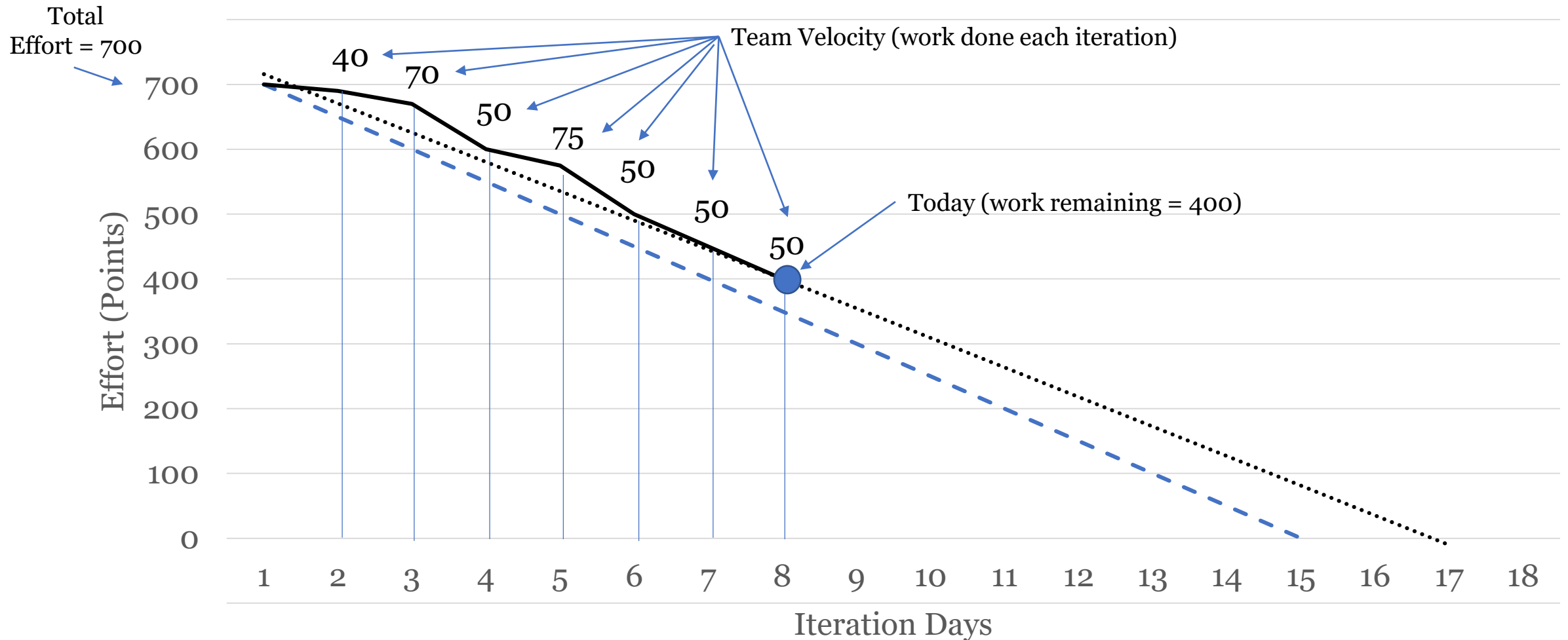
# Iteration Burndown Chart

- This is a chart showing actual remaining work, ideal remaining work, forecasted remaining work, and velocity.
- It is used to measure and predict variance between the ideal and actual burndown.
- A point system is typically used to represent the amount of work that must be performed (e.g., 1,000 points are needed to complete the endeavor).

# Iteration Burndown Chart Terms

- **Actual Remaining Work:** this metric illustrates how much effort (work) has been expended to date and, therefore, how much effort (work) remains.
- **Ideal Remaining Work:** this metric illustrates how much work was originally planned to finish the iteration.
- **Forecasted Remaining Work:** this metric extrapolates how much work will likely be required to finish the iteration (a linear regression model based on previous actual performance).
- **Velocity:** the amount of points earned in each iteration. It illustrates how much work was performed.

# Iteration Burndown Chart



-- Ideal Remaining Work    — Actual Remaining Work    ..... Forecasted Remaining Work

**Figure 6-67: Iteration Burndown Chart**



# Time Management Summary

- As this presentation illustrates, managing time in a project is essentially about creating or adopting, implementing, and managing a project schedule.

# Activities and Resources

- Time management activities include the creation or adoption and modification of a schedule management plan for a project. We examined methods for identifying of activities and milestones associated with a project.
- In addition, we identified the key attributes associated with each activity to include the resources that are needed to complete the activities. The complexity of a schedule management plan varies and is based on the intricacies of a project.

# Activities and Sequencing

- We examined how to sequence activities using precedence diagramming methods. We then observed how to calculate the durations of the activities using analogous, parametric, and three-point estimates and the program evaluation and review technique among others.

# Schedule

- We learned how to create a schedule and determine critical pathways by performing forward and backward passes and float calculations.
- We then examined how to adjust the schedule as needed during its execution by applying leads and lags, conducting what-if analysis (e.g., Monte Carlo simulations), and other approaches.
- Again, by employing these various resources, our aim is the creation or adoption, implementation, and management of a schedule.

# Questions?