All Online Learning

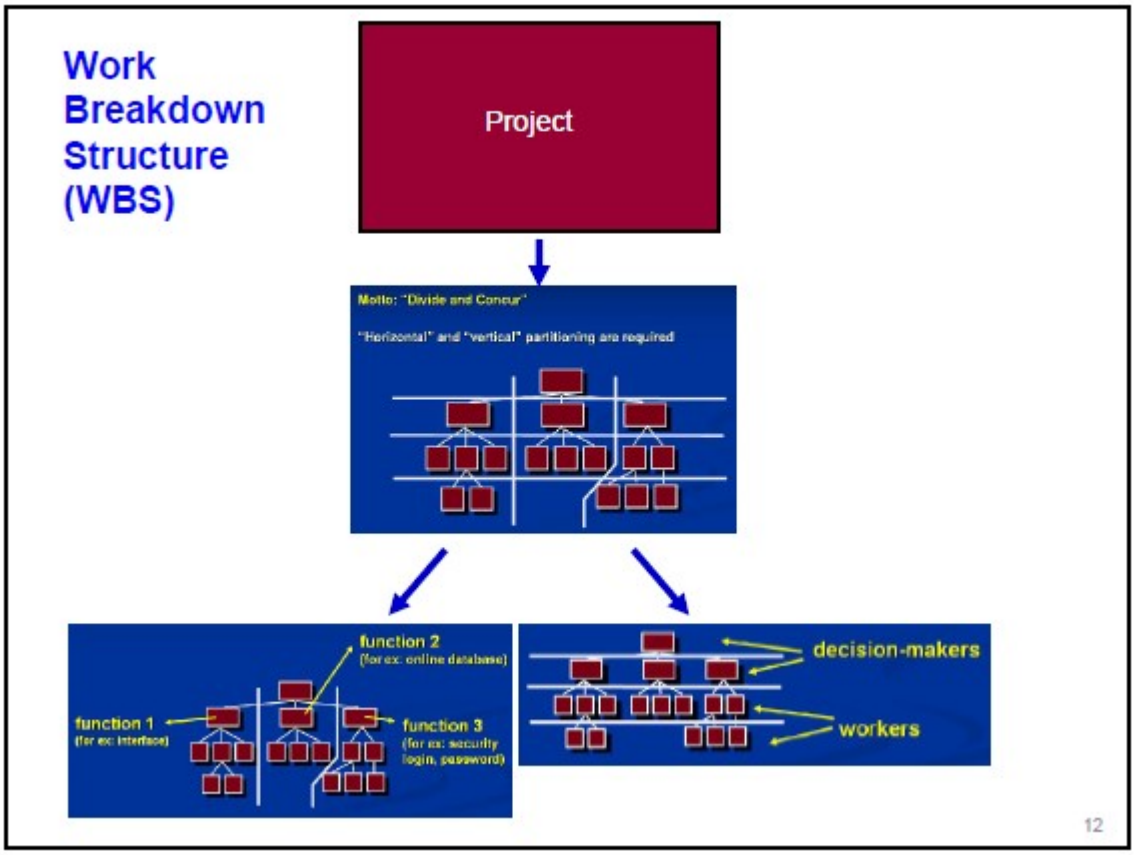
www.allonlinelearning.com

Unit 2

Lecture 1: Project Elements, Work Breakdown Structure (WBS) Types of WBS, function activities and tasks

- A work breakdown structure (**WBS)** is a method used to define group of project's discrete work elements (or, tasks) in a way that helps organize and define the total work scope of the project.
- **WBS element** may be a product, data, a component, a service, or any combination.

- **100% rule:** The WBS represents 100 percent of the work required to produce the final products, and, therefore, all tasks must add up to 100% of the total scope and should not



Building the WBS:

Various approaches can be used to build the WBS:

1. **The analogy approach:** A WBS is first created by looking for a similar projects done in the past and using its WBS as a starting point. SE Design Concept: "Do NOT reinvent the wheel" (check web sites of similar projects)
2. The top-down approach: Start with the largest items of the project and keep breaking them down into smaller and smaller parts
3. The bottom-up approach: Start with the detailed tasks and roll them up
4. Thread-based approach: Concentrate on most important items first

Analogy Approach: Advantages and Issues

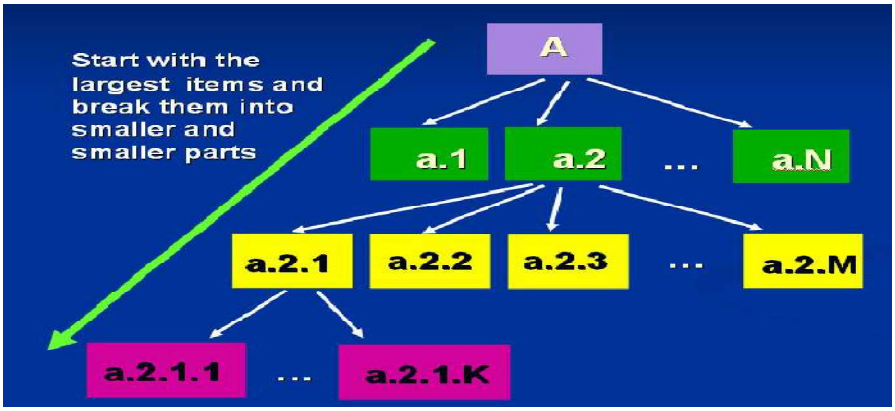
Advantages:

- Is the fastest path to a completed WBS
- Is a valuable tool for brainstorming a new project and looking for deliverables
- Enhances cross-project consistency
- Improves budget and time estimates
- Improves resource allocations

Issues:

- Ensure that the previous WBS is completely understood and similar
- Ensure the previous WBS is accurate and updated
- Critically review the previous WBS and its appropriateness for the new project

Top-Down Approach: Advantages and Issues



Advantages:

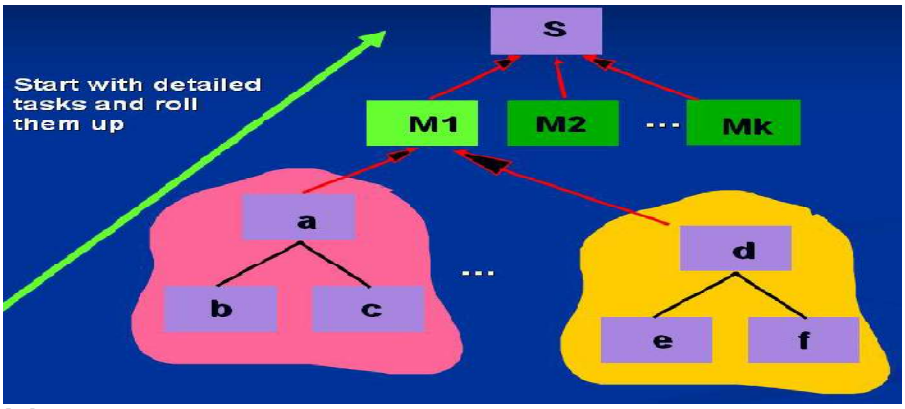
- Ensures projects are organized logically based on the nature of the project
- Promotes stakeholder participation in the planning phase of the project
- Can create a greater understanding of the entire project by all participants

Issues:

- Need to make sure major objectives are not forgotten
- Make sure to decompose the tasks to appropriate levels
- Can be time consuming, must guard against "analysis paralysis"

- Cost and time estimates are more difficult to create and generally less accurate than under the analogy approach

Bottom-Up Approach: Advantages and Issues



Advantages

- May lead to a more complete list of tasks and detailed description of tasks
- Promotes participation of various stakeholders in the planning phase of the project
- Can create a greater understanding of the entire project by all participants

Issues

- Difficult to retain focus on the “big picture”
- Need to make sure major objectives are not forgotten
- Harder to get organized into logical steps or phase

Thread Approach: Advantages and Issues

Advantages

- Generally the most important stakeholder objectives done first
- Greater control and focus of the brainstorming sessions
- Promotes stakeholder participation in the planning phase of the project
- Can create a greater understanding of the entire project by all participants

Issues

- Make sure to not lose focus of the “big picture”
- May lose site of the effect one objective may have on another
- Increases the need for communication
- More successful when the project leader and team has a good understanding of the project’s objectives

Which Approach to Choose?

In case of the existence of a similar project:

Would lead you to the analogy approach which if done correctly is the fastest and most accurate method

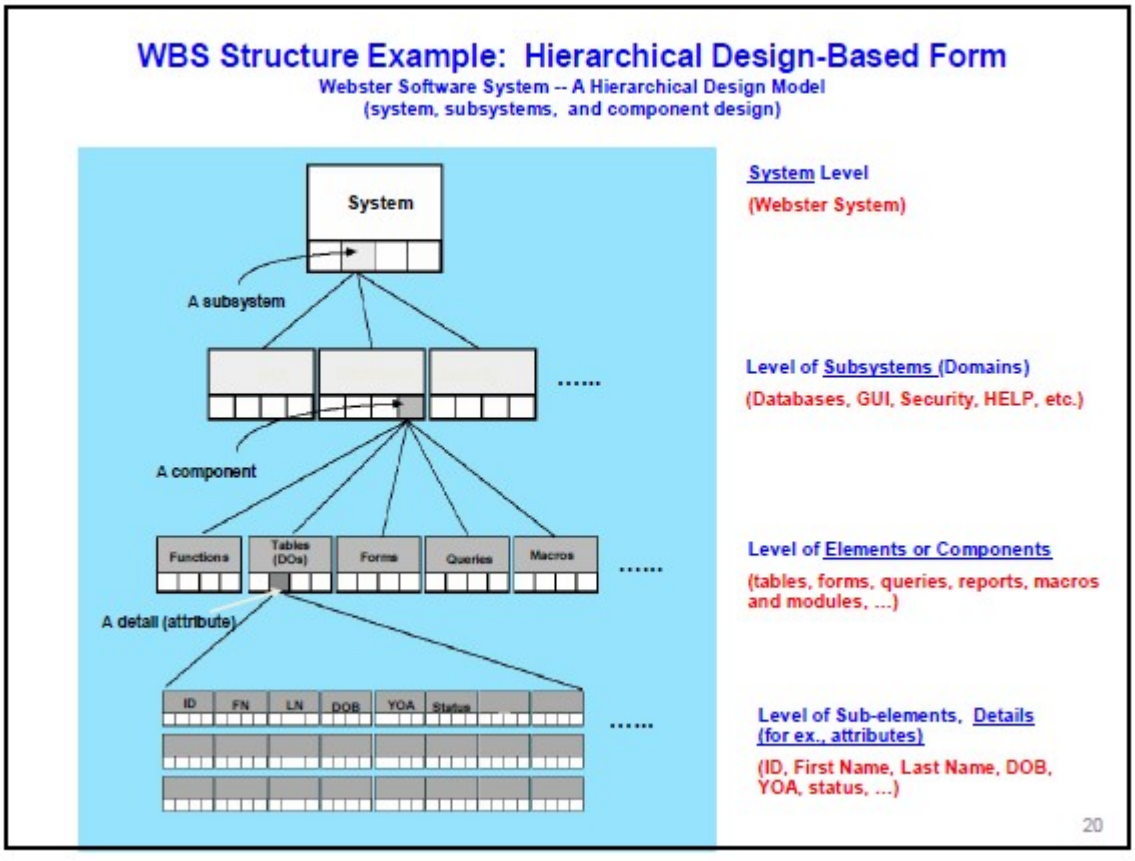
In case of an evolutionary type of project:

Depends on experience level of the project manager and team:
- if little experience, choose the top-down approach;
- if many years of experience then choose a bottom-up approach

In case of a revolutionary type of project:

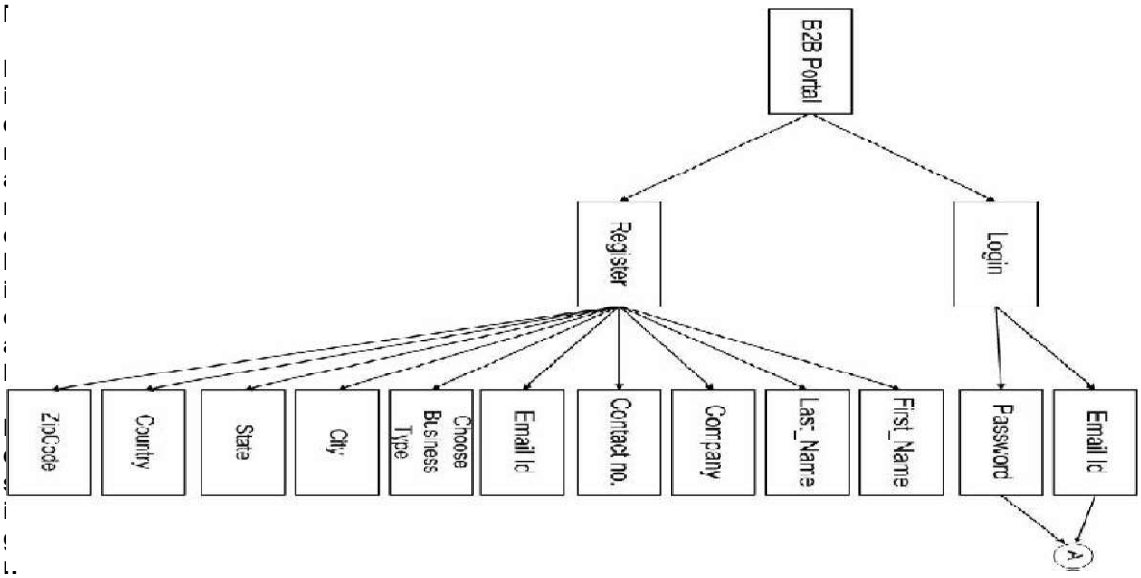
- if the product or process is very unique, never anything like it before in this company or by this team then choose the top-down approach

WBS Structure Example: Hierarchical Design-Based Form

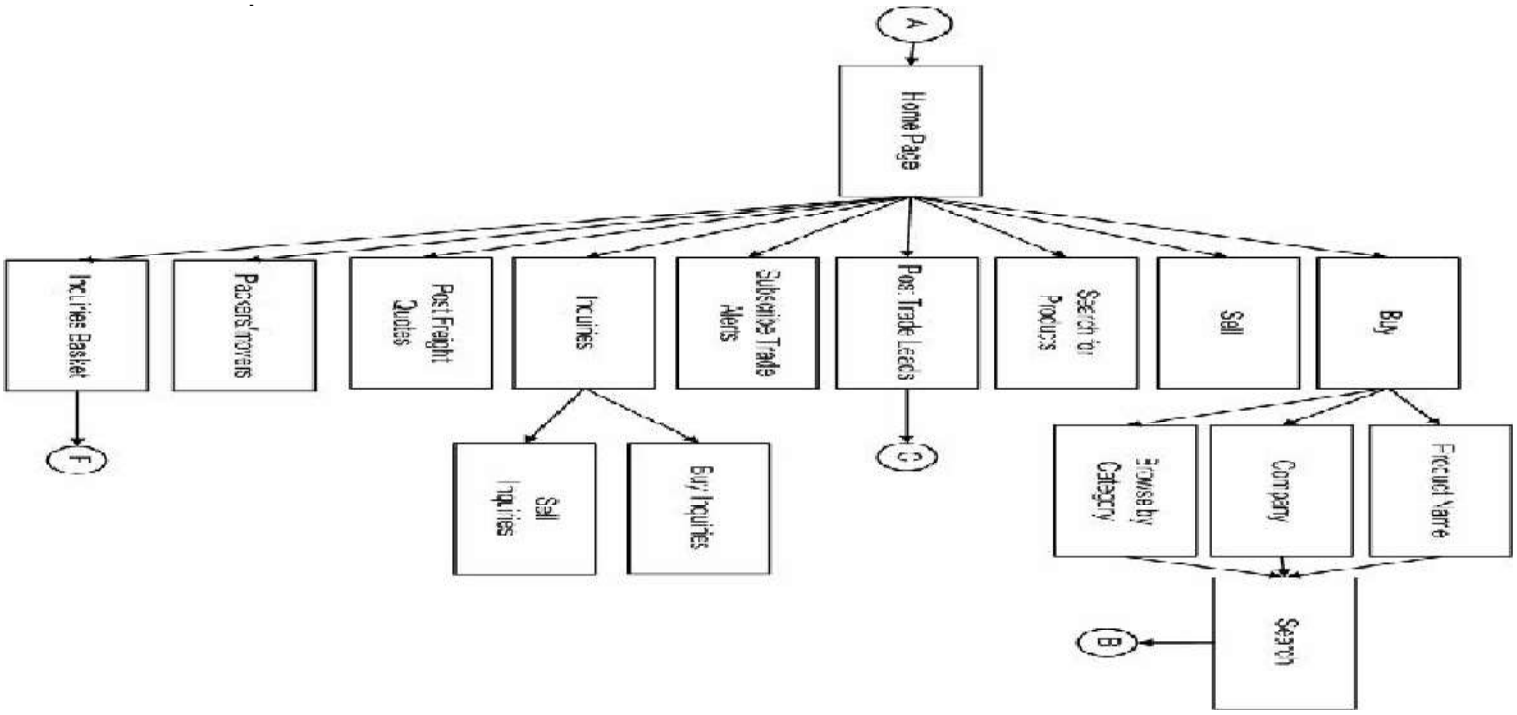


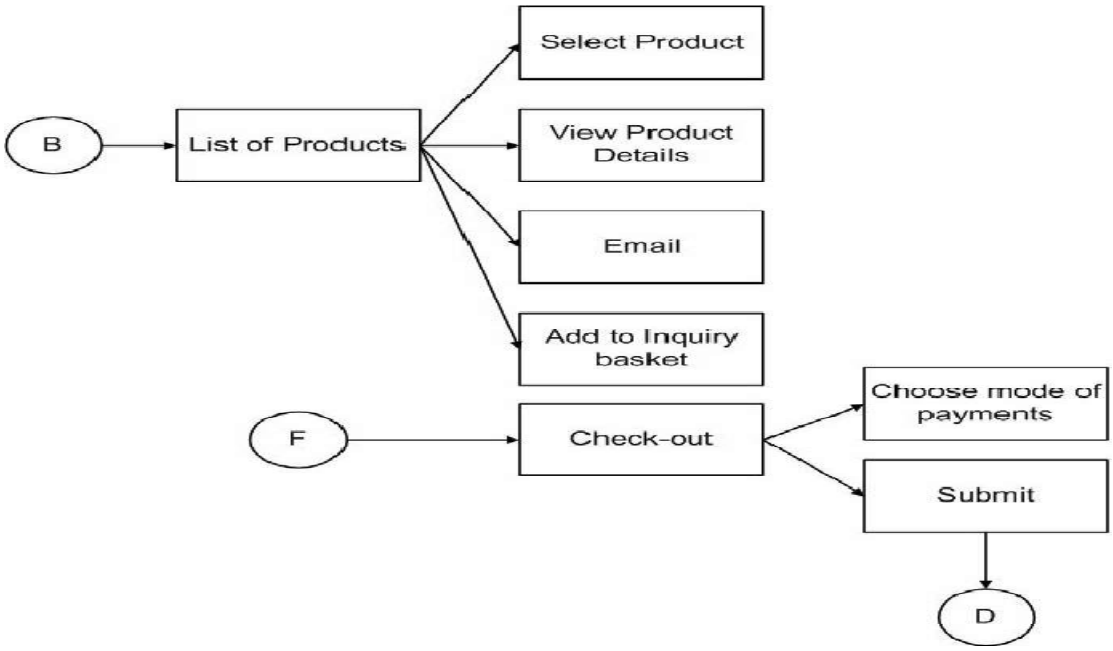
WBS Example: Decision Tree-Based Form

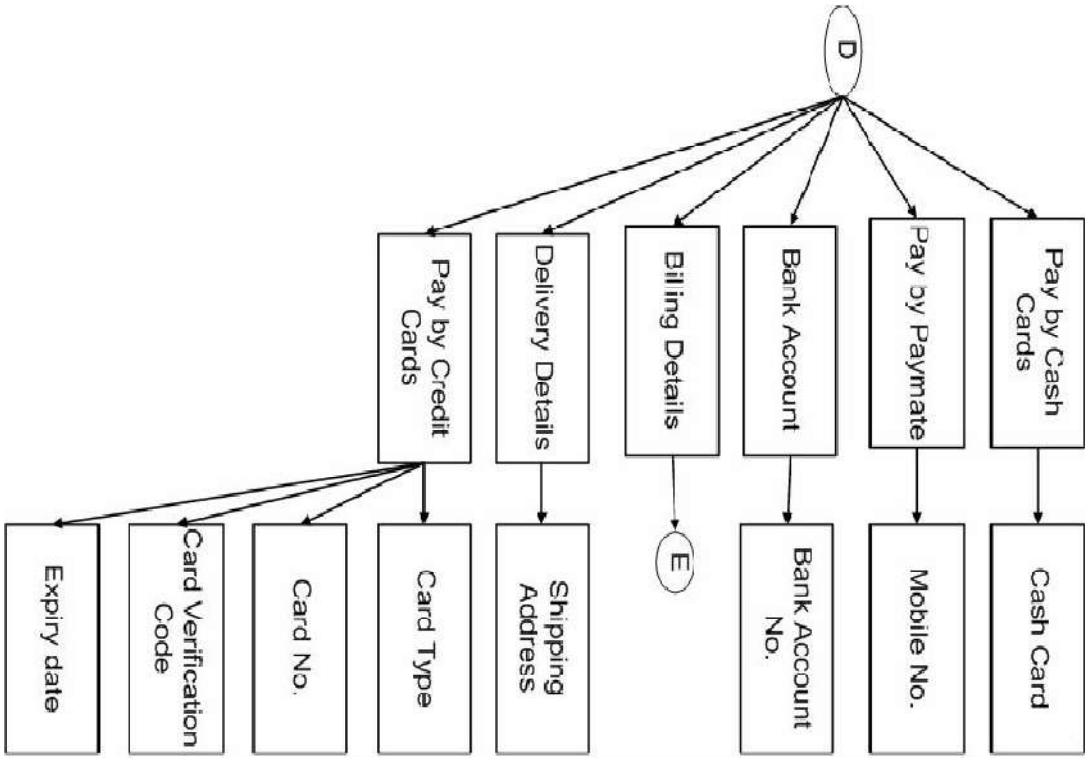
Example: GUI



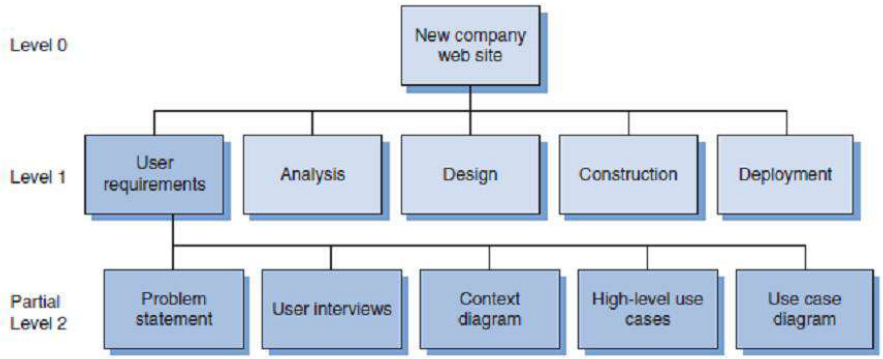
Mo

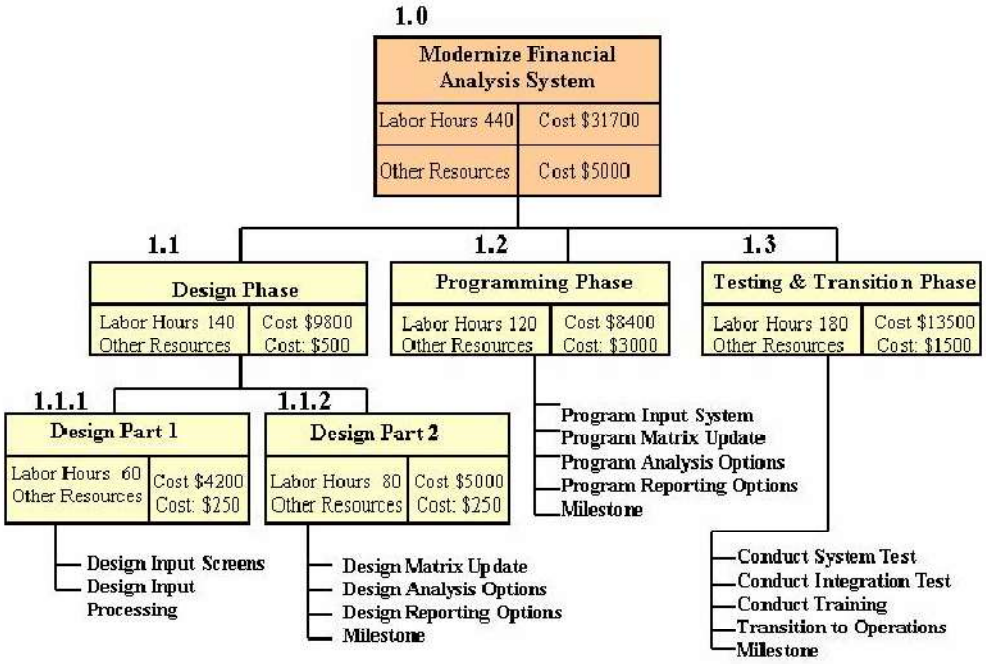



B
List of Products
Select Product
View Product Details
Email
Add to Inquiry basket
F
Check-out
Choose mode of payments
Submit
D



WBS Examples: Process-Based Form




1.0
Modernize Financial Analysis System
Labor Hours 440
Cost $31700
Other Resources
Cost $5000
1.1
Design Phase
Labor Hours 140
Cost $9800
Other Resources
Cost: $500
1.2
Programming Phase
Labor Hours 120
Cost $8400
Other Resources
Cost: $3000
1.3
Testing & Transition Phase
Labor Hours 180
Cost $13500
Other Resources
Cost: $1500
1.1.1
Design Part 1
Labor Hours 60
Cost $4200
Other Resources
Cost: $250
1.1.2
Design Part 2
Labor Hours 80
Cost $5000
Other Resources
Cost: $250
Program Input System
Program Matrix Update
Program Analysis Options
Program Reporting Options
Milestone
Design Input Screens
Design Input Processing
Design Matrix Update
Design Analysis Options
Design Reporting Options
Milestone
Conduct System Test
Conduct Integration Test
Conduct Training
Transition to Operations
Milestone

WBS Example: Tabular Form

		Normal	Predecessors
1	**Project Initiation**		
1.1	Develop project charter	5 days	
1.2	Develop Statement Of Work	8days	1.1
1.3	Develop preliminary scope development	2days	1.2
1.4	Develop preliminary architectural model	5days	1.3
1.5	Project initiation complete	2days	1.4
2	**Project plan**		
2.1	Develop scope management plan	2days	1
2.2	Develop change management plan	5days	2.1
2.3	Develop intial descriptive budget	10days	2.1, 2.2
2.4	Develop schedule	3days	2.2, 2.3
2.5	Develop quality management plan	4days	2.4
2.6	Develop human resource plan	3days	2.5
2.7	Develop risk management plan	1 day	1
2.8	Project paln complete	1 day	2.1 - 2.7
3	**Project Execution**		
3.1	Release 1		
3.1.1	Analysis phase	15days	2
3.1.2	Design phase	10days	3.1.1
3.1.3	Construction phase	8days	3.1.2
3.1.4	Validation phase	15days	3.1.3
3.1.5	Deployment phase	3days	3.1.4
3.1.6	Closeout	1day	3.1.5
3.1.7	Release 1 Complete	2days	3.1.6
3.2	Release 2		
3.2.1	Analysis phase	15days	3
3.2.2	Design phase	10days	3.2.1
3.2.3	Construction phase	8days	3.2.2
3.2.4	Validation phase	15days	3.2.3
3.2.5	Deployment phase	3days	3.2.4
3.2.6	Closeout	1day	3.2.5
3.2.7	Release 2 Complete	2days	3.2.6
3.3	execution complete		
4	**project closeout**	2days	3
5	**project complete**	1days	4

Basic Principles for Creating a WBS

- The WBS represents 100% of the work required to produce the product. As soon as you define more than 100% of the scope, you have committed to doing more than you agreed to - scope creep has begun (100% Rule)
- Each WBS element represents a single deliverable
- Each deliverable is distinct
- Accountability for each task can be assigned to one team member
- Not all elements of the WBS need to be decomposed to the same depth
- Have all reporting and control mechanisms been included
- Be prepared for changes

Dictionary:

- Control accounts – accounting or finance department assigned account codes used in the accounting system to track costs
- Statement of work – describing the details of the work involved in creating each deliverable
- Responsible organization – who is responsible for each deliverable

- Schedule for major milestones
- Contract information if outside vendor involved
- Quality requirements
- Estimate of cost and resources required

Project Plans: Elements

Firstly, I need to make sure we are all on the same page when it comes to what a plan is. Many people (and a distressing number of project managers, too) think only of a Gantt chart when they think of a project plan. You may recognize it as what you get from Microsoft Project. This is better called a project schedule, in that it shows when we expect the various sections of the project to happen. We will come back to this later.

What we want to have in our project plan is:

1. Aim of Project
2. Outputs
3. Quality Criteria
4. Resources
5. Management Structure
6. Milestones
7. Tolerances
8. Dependencies
9. Risks
10. Schedule

Let's have a look at these in turn, and see why they are needed, and what we want to achieve with each of them.

1. Aim of Project

What do we want to produce? The aim of the project is a mixture of the reasons for doing the project and the benefits that are expected from it. This section of the plan can be either fulfilled by linking to the main business case, or by restating it in language for the expected audience. For example, your business case may have been written for high level approval in your organization. You may want to now put it in terms the project executive expects.

2. Outputs

Given the aim of the project, what do we actually need to produce to get there? What will your completed project be made up of? These need to be clearly defined. For example, your project's aim may be to upgrade the IT infrastructure in an organization. Your final output would be a completed computer network, a new computer on every desk, and all appropriate software installed and ready to go.

3. Quality Criteria

Now we have the outputs, we need to understand what quality they need to be of. In the example above, we have an output of a completed computer network. However, we need to know that the network can actually cope with the amount of traffic going over it!

This means we need the completed output to be of a certain quality, and we need to define what that quality is. These targets tell you what success is, what completion of the project is. They need to be SMART:

- **Specific:** Clearly defined and precise.
- **Measurable:** e.g. not "new computers," but "computers with 8GB of memory", etc.
- **Attainable:** Don't ask for the impossible.
- **Relevant:** Is the criterion actually related to the aim of the project?
- **Time-based:** Enough time to achieve this. There is no point expecting a year's worth of work in one week!

It is important you take some time with the stakeholders in your project to produce this list. The final customer of the project will naturally be very involved, but don't forget your business head - don't promise everything without considering the costs. Your project executive, and a representative of those who will be doing the work, will have major inputs into this also.

Finally, you will also need to decide who has the final say over the quality of the outputs. Hopefully your work on defining the quality criteria will mean there are no arguments over the quality (i.e. no qualitative judgments, only quantitative) but you need to make sure you schedule in time and people to do the evaluation work.

4. Resources

We have now set down what outputs we need to produce, and what quality they need to be at. This means we are now in a position to look at the resources we will need to achieve this. Resources include staff time, particular knowledge or skill sets, money (e.g. buying equipment), and time (some tasks can't be increased by throwing more people at the problem, e.g. delivery times, setting time for concrete, etc.).

5. Management Structure

How are we going to manage the work? You need to describe the general approach to the project here. Who will be the decision makers for the various different streams of work? For example, you may be doing a significant procurement - who makes the decision about what company to buy from?

How will we share progress on the project? Who will we share it to? For example, you may decide to have regular project team meetings - who needs to attend? What level of information will you be sharing? Who else needs to be kept informed, at what level of detail, and how often? For example, you may want to keep the project executive updated at an overview level of detail on a weekly basis, while you keep other managers appraised at a higher level of detail.

You will also need to spell out the relationship of yourself to the project executive, in terms of the monitoring of progress. Equally, you need to put down how you will be monitoring progress of the allocated tasks.

There is no one right answer for how this should be done, and indeed it will vary with every project. Make sure you think about the size and complexity of the project, and also the organisation's ethos and current management style.

6. Milestones

Here you need to think about how you will break up the project. Unless it is very small, you don't want to have the entire project as one lump of work, with the only check on progress at the very end. Instead, it makes sense to break the project up into discrete chunks, where related tasks can be lumped together, with a sensible milestone at the end of them. For example, in the technology refresh in the example above, you may want to break the project down into something like:

1. Requirements Gathering
2. Tender Writing
3. Tendering Process
4. Contract Negotiation
5. Deployment
6. Testing

It makes sense to have a defined milestone, so you know when each section is completed. There is also another benefit of breaking the project into chunks, which I'll come back to.

7. Tolerances

You will have already looked at the resources you need. Now we need to set how far you, or the project executive, can let the project stray from these targets before needing to sound the alarm. For example, you could set a tolerance in terms of finance of +/- 5%, and a tolerance in terms of time of +/- 10%. Equally, you may want to look at tolerances of quality - i.e. how far from the quality criteria are you willing to accept?

It is remarkably unlikely that a project will not deviate from its resource or quality targets. Setting tolerances allows you to be able to manage the project without continually seeking guidance from the project executive as to whether you should carry on. This is not to say that you should be happy with these deviations, and you should try to avoid them, and monitor them closely. That way you can build your understanding of the project for the future.

8. Dependencies

This is where you look at what needs to happen before something else. For example, in our example above, you need to complete the requirements gathering before you can finish the tender documentation. You need to start thinking about the dependencies so you, and the project team, can understand the impact of changes in any part of the project.

These dependencies should include both those internal to the project (i.e. those under your control), and those external to it (i.e. those outside of your control). For example, you may need an accurate figure for the number of staff in the organization. This needs to come from your HR department, and would be an external dependency.

9. Risks

Simply, what could go wrong? What could happen that would damage your ability to deliver the project? Are there things you can do to avoid them, or minimize them?

10. Scheduling

This is the Gantt chart-style information that many people envisage when a project plan is mentioned. In this, you need to put down what you expect to happen when. It will include your dependencies, milestones, and probably resources. At this point, it will be a relatively high overview of the whole project.

There is something you need to understand about this schedule, and that is this: it will be wrong.

I know that seems a strong statement, but it is vital that you understand that you cannot make a perfect schedule. You really would be getting into the realm of prophecy if you think you can sit down now, and accurately and precisely pinpoint the date the project will end. No, what you need to do here is achieve the possible.

The schedule needs to include the overview, with the project broken down into sensible chunks. This is the other advantage of breaking the project into chunks we mentioned above. By having the project broken up in this way, you will be able to start planning the first section in quite some detail, extending out for a few weeks. But from then on, it will start to be based more and more on blind guesswork and faith. Don't try to be artificially precise - keep it vague, use rough figures.

As you come to the end of each chunk of the project, you will be able to plan the next one. You can use the information and experience you have just gained from the previous section, and thus you will be able to be more confident.

Make sure you explain this to your project stakeholders! Often your project executive may look at a schedule, and imagine everything is laid out and known. You must get this idea out of their head straight away! Explain that the early part is firmer than the rest, and make sure they expect changes as the project moves on.

Your executive will crave certainty, and absolute dates for the project, from the very beginning. You must resist the pressure to name a specific date, and explain why. While there may be a temptation to give an answer (no doubt of a date plucked, essentially, from the air) you need to instead be realistic about what is and isn't possible in terms of scheduling. Anything else can only lead to trouble for you, the project, and ultimately your executive further down the line.

Lecture 2. Project Life Cycle and Product Life Cycle, Ways to Organize Personnel

For many non-project managers it is often difficult to understand the full extent of the role of the Project Manager throughout a project. The various stakeholders tend to view their projects in terms of a sequence of steps that can be easily represented in a Gantt chart often called the Project Plan – more on the use of that term later. While tasks and activities on the Gantt chart represent work that needs to be done at a particular moment in time, there is a range of management skills and processes that are applied on an ongoing basis throughout the project and, as such, are not represented on the chart as specific activities occurring at particular moments in time.

This article sets out to provide greater clarity as to what the Project Manager's role is by decomposing the work in a project into two streams
- the product and the
-project management lifecycles.

The Product lifecycle has a defined approach, depending on the type of project, utilizing phases and templates to direct the sequence and deliverables associated with the project.

The lifecycle would be different for a software implementation from, say, that of building a ship.

Project Management refers to the application of the professional body of knowledge combined with personal skills a project manager brings to the management of the project. These knowledge and skills are essentially the same irrespective of the type of project. Both of these components, or streams of the project lifecycle, are integrated to form the essential framework for the effective delivery of the project.

Product Lifecycle

The Product or Delivery Lifecycle is composed of a series of phases which are a logical and sequential grouping of the activities that are required to be undertaken to deliver the end product.

These activities may be undertaken in a **waterfall** type manner, that is, a preceding activity must be performed before the next activity is started. As an alternative these activities can be performed in an **iterative** manner, going through a cycle of activities, e.g. prototyping, to facilitate the continuous improvement of the end product. Another approach is to deliver in an **incremental** manner – deliver quick wins.

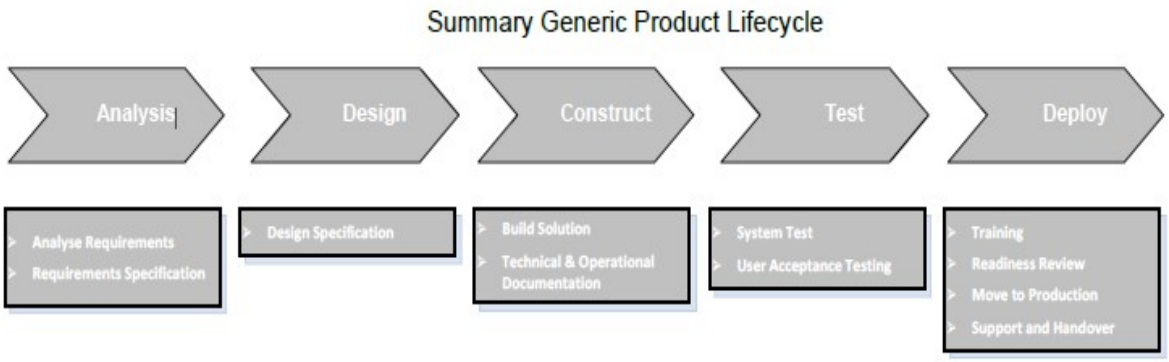
The traditional product delivery "waterfall" lifecycle goes through a number of phases.

These include

(i) requirements analysis,
(ii) design,
(iii) construct,
(iv) test and
(v) Deployment.

In addition to these phases, before the project formally commences and after closing, there will most likely be a start-up phase, that is, justifying and formally starting any project, and an operational phase, involving the day to day management of the functioning product at the end of the project.

The following diagram illustrates the sequence of Product lifecycle phases together with a number of the key activities:



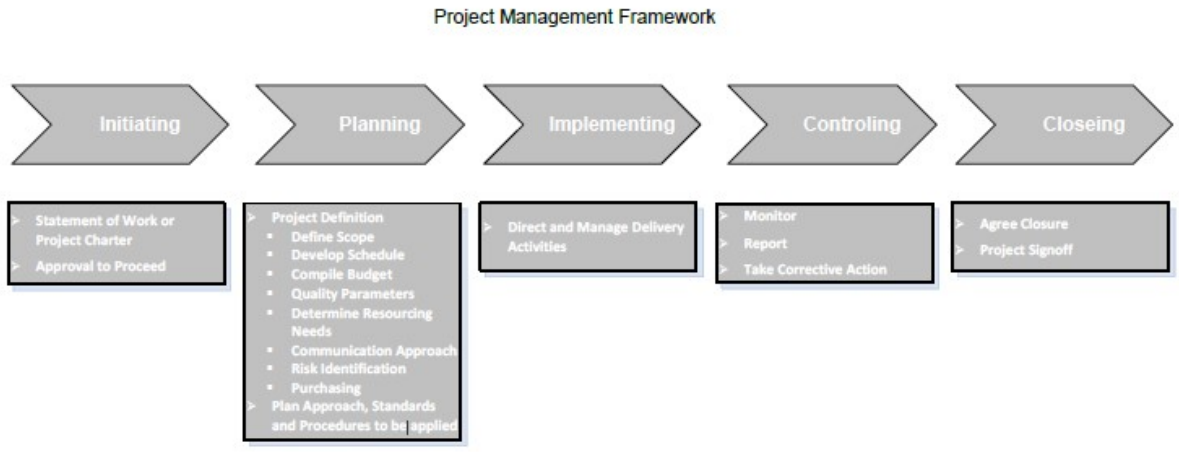
Project Management Cycle

The knowledge, skills, tools and techniques that are applied to a project by a Project Manager follow a different, but complimentary cycle and the associated processes are categorized under Scope, Quality, Procurement, Cost, Time, Resources, Risk/Issues, Stakeholder and Communications Management.

These processes are phased under various process groups:

- **Initiation –** The processes involved in converting the project vision into a definition of the expectations of the stakeholders and providing the formal handover to the project manager
- **Planning –** The processes involved in the develop of a detailed definition of the work and as to how it is to be executed and managed
- **Implementing –** Managing the work based on the product lifecycle work plan
- **Controlling –** Track, review and regulate the progress and performance of the project
- **Closing –** Finalize and handover the project output

On receiving formal approval to commence the project, the Project Manager will set about determining the scope of the project, not just in term of with the final output is to be, but also all the parameters associated with managing the project to achieve that output.



The Project Manager will define sequence, determine the effort and schedule the activities to be undertaken throughout the project and determine who will undertake these activities. Drawing up and agreeing the budget, identifying the risks and determining the appropriate way to communicate with all stakeholders are included in the planning activities to be undertaken by the Project Manager.

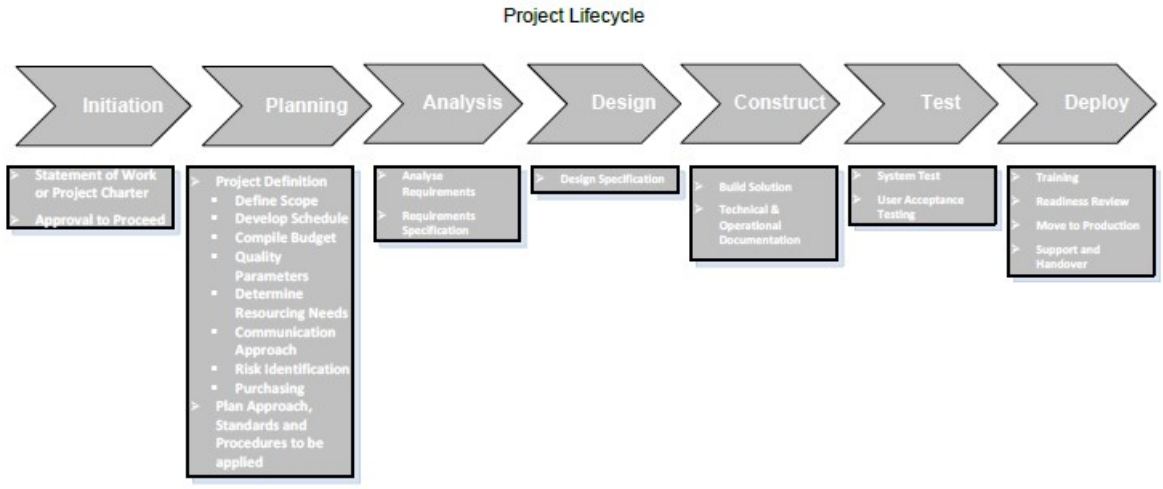
The planning activities will involve determining what standards and procedures will be applied throughout the project in terms of risk, issues, change control, schedule, costs, communications, configuration management and quality. Having completed and signed-off the Planning phase, the Project Manager will be responsible for managing the performance of the tasks and activities associated with the Product lifecycle. Part of the responsibility will include monitoring and controlling the work and taking corrective action where appropriate.

Finally at the conclusion of the project the Project Manager will oversee the formal closure of the project.

Integrating Product Lifecycle and Project Management Processes

Having decomposed the components of the project lifecycle it is now appropriate to bring them together to reflect the overall sequence with the full visibility of the Project Manager's involvement.

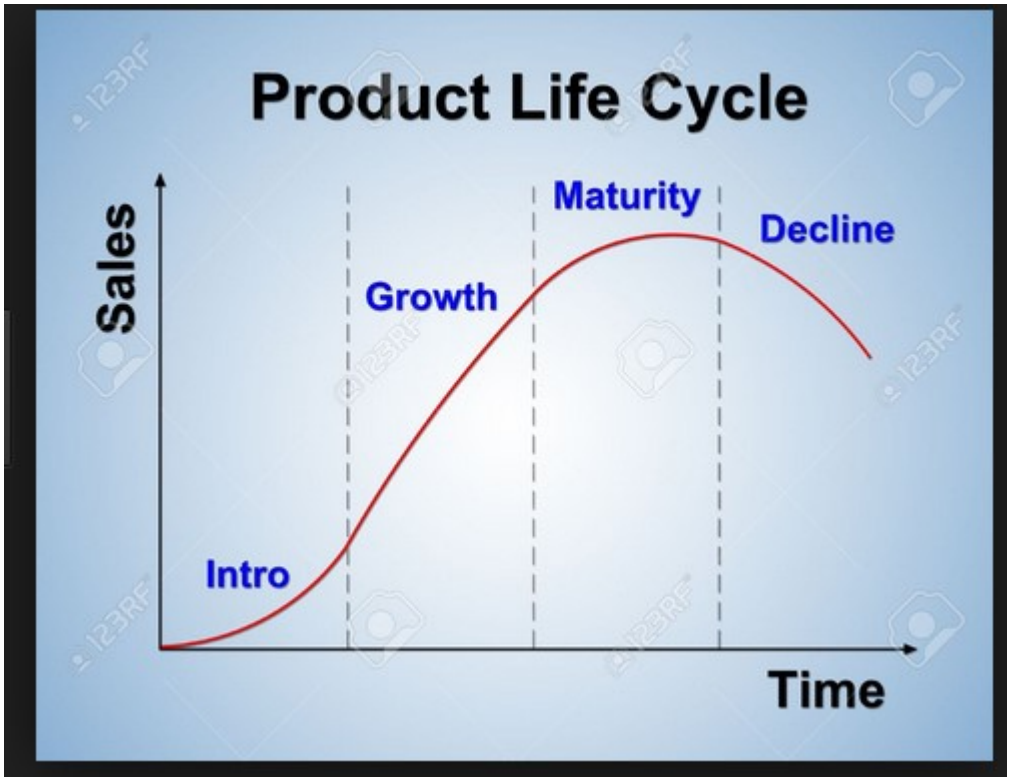
An organization's or industry specific methodology is developed by bringing together all the elements of the Product lifecycle and Project Management framework and setting out in detail the inputs and outputs and the tools and techniques used through the project put from a scheduling tool is a work plan



Product Life Cycle Stages

As consumers, we buy millions of products every year. And just like us, these products have a life cycle. Older, long-established products eventually become less popular, while in contrast, the demand for new, more modern goods usually increases quite rapidly after they are launched.

Because most companies understand the different product life cycle stages, and that the products they sell all have a limited lifespan, the majority of them will invest heavily in new product development in order to make sure that their businesses continue to grow.



Product Life Cycle Stages Explained

The product life cycle has 4 very clearly defined stages, each with its own characteristics that mean different things for business that are trying to manage the life cycle of their particular products.

Introduction Stage – This stage of the cycle could be the most expensive for a company launching a new product. The size of the market for the product is small, which means sales are low, although they will be increasing. On the other hand, the cost of things like research and development, consumer testing, and the marketing needed to launch the product can be very high, especially if it's a competitive sector.

Growth Stage – The growth stage is typically characterized by a strong growth in sales and profits, and because the company can start to benefit from economies of scale in production, the profit margins, as well as the overall amount of profit, will increase. This makes it possible for businesses to invest more money in the promotional activity to maximize the potential of this growth stage.

Maturity Stage – During the maturity stage, the product is established and the aim for the manufacturer is now to maintain the market share they have built up. This is probably the most competitive time for most products and businesses need to invest wisely in any marketing they undertake. They also need to consider any product modifications or improvements to the production process which might give them a competitive advantage.

Decline Stage – Eventually, the market for a product will start to shrink, and this is what's known as the decline stage. This shrinkage could be due to the market becoming saturated (i.e. all the customers who will buy the product have already purchased it), or because the consumers are switching to a different type of product. While this decline may be inevitable, it may still be possible for companies to make some profit by switching to less-expensive production methods and cheaper markets.

Product Life Cycle Examples

It's possible to provide examples of various products to illustrate the different stages of the product life cycle more clearly. Here is the example of watching recorded television and the various stages of each method:

1. Introduction – 3D TVs
2. Growth – Blueray discs/DVR
3. Maturity – DVD
4. Decline – Video cassette

The idea of the product life cycle has been around for some time, and it is an important principle manufacturers need to understand in order to make a profit and stay in business.

However, the key to successful manufacturing is not just understanding this life cycle, but also proactively managing products throughout their lifetime, applying the appropriate resources and sales and marketing strategies, depending on what stage products are at in the cycle.

Way to organize personnel:

Organizational configurations:

Five ways organizations typically configure and coordinate teams:

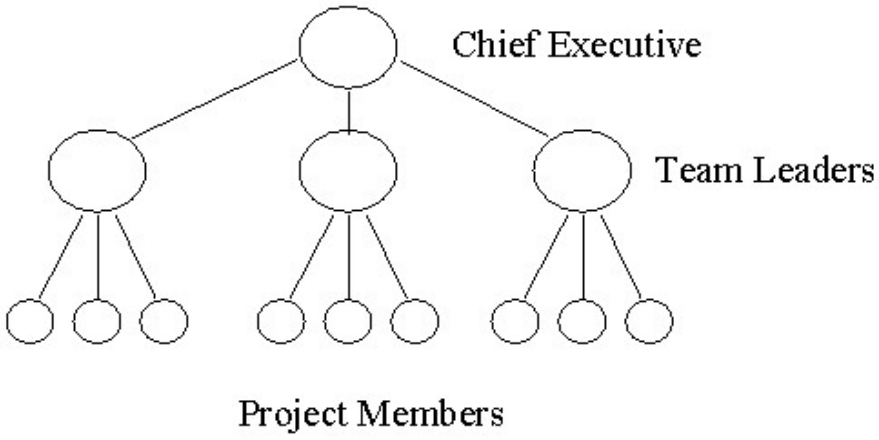
- Simple structure – one or few managers, direct supervision
    - Typically found in new, relatively small organizations
- Machine bureaucracy – mass-production and assembly lines
    - Coordination requires standardization of work processes
- Divisionalized form – each division has autonomy
    - Split up work and let each group figure out how to do it
    - Coordination achieved through standardization of work outputs and measuring performance of divisions
- Professional bureaucracy – skilled professionals with autonomy

- Coordination achieved through standardization of worker skills

- Adhocracy – for innovative or exploratory projects
  - Coordination achieved through mutual adjustment

Which configurations apply for software development projects?

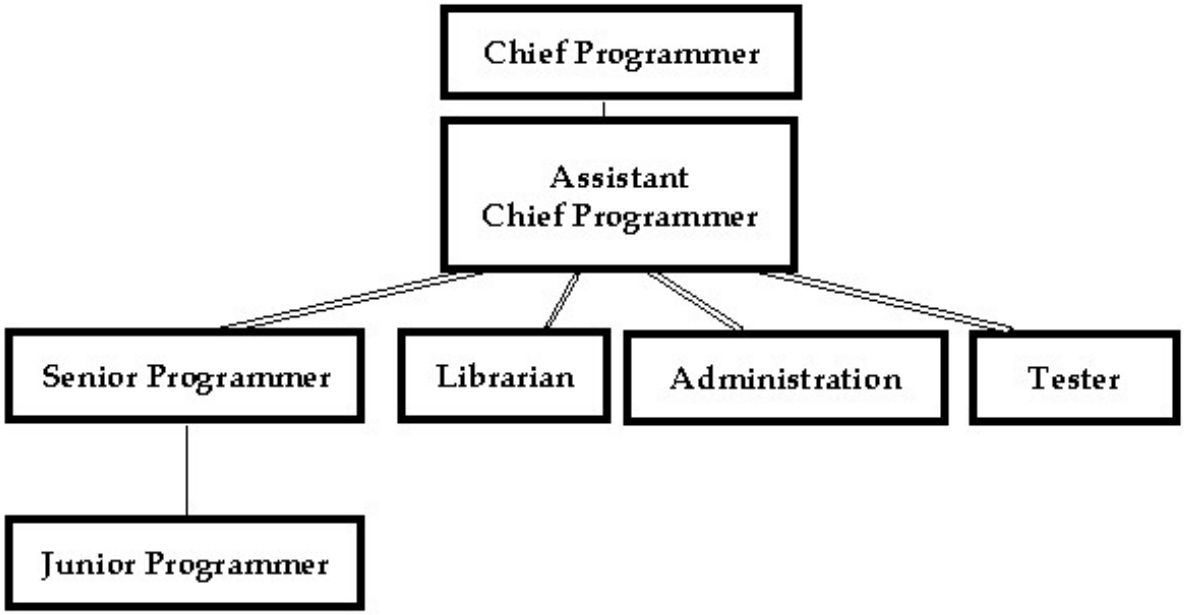
Hierarchical team organization:



Large projects often distinguish levels of management:

- Leaf nodes is where most development gets done; rest of tree is management
- Different levels do different kinds of work—a good programmer may not be a good manager
- Status and rewards depend on your level in the organization
- Works well when projects have high degree of certainty, stability and repetition
- But tends to produce overly positive reports on project progress, e.g.:
  - Bottom level: “We are having severe trouble implementing module X.”
  - Level 1: “There are some problems with module X.”
  - Level 2: “Progress is steady; I do not foresee any real problems.”
  - Top: “Everything is proceeding according to our plan.”

Chief Programmer Team:



What do the graphics imply about this team structure?

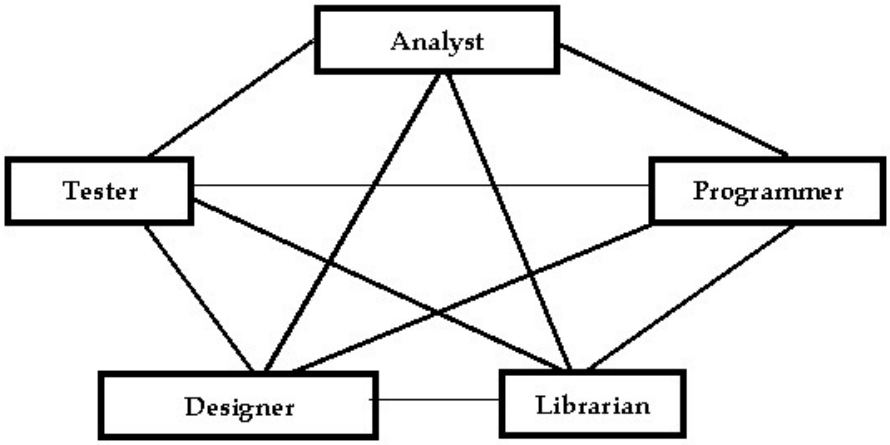
- Chief programmer makes all important decisions
    - Must be an expert analyst and architect, and a strong leader
- Assistant Chief Programmer can stand in for chief, if needed
- Librarian takes care of administration and documentation
- Additional developers have specialized roles
- *Pros and cons of this team structure? Will you use this organization?*

Matrix organization:

	real-time programming	graphics	databases	QA	testing
project C	X			X	X
project B	X		X	X	X
project A		X	X	X	X

- Organize people in terms of specialties
    - Matrix of projects and specialist groups
    - People from different departments allocated to software development, possibly part-time
- *Pros and cons?*
    - Project structure may not match organizational structure
    - Individuals have multiple bosses
    - Difficult to control a project's progress

Democratic or Open structured teams:



Why are democratic teams often favoredin Extreme Programming process?

- A "grass roots" anti-elitist style of team organization
    - Egoless: group owns the documents & code (not individuals)
    - All decisions are based on team consensus
    - Depends on total cooperation of its members
    - Requires clear structure for the way the team interacts
    - Functional roles (e.g. moderator, recorder) rotate among team members
    - A technical leader has external responsibility and resolves issues when team doesn't reach consensus

Lecture 3: Project schedule, Scheduling Objectives, Building the project schedule, Building the project schedule, Scheduling terminology and techniques,

Project schedule:

- Def.: A schedule is the conversion of a project action plan into an operating timetable
- it Is the basis for monitoring and controlling project activity
- Taken together with the plan and budget, it is probably the major tool for the management of projects
- In a project environment, the scheduling function is more important than it would be in an ongoing operation
- Projects lack the continuity of day-to-day operations (routine) and often present much more complex problems of coordination
- The basic approach of all scheduling techniques is to form a network of activity and event relationships
- This network should graphically portray the sequential relations between the tasks in a project
- Tasks that must precede or follow other tasks are then clearly identified, in time as well as function

Building project schedule:

How to Build a Project Schedule in 5 Easy Steps

Congratulations! You've been assigned your first project and your boss wants to see a project schedule at next week's status meeting. Hearing of your new promotion, the PC support team has installed Microsoft Project on your desktop so you're ready to start building a project schedule! Unfortunately, your Introduction to Microsoft Project training class isn't schedule until next month and your boss is expecting a full schedule by next week. Fortunately, if you have a book on Microsoft Project and this article, you'll be able to complete your assigned task.

According to the Project Management Body of Knowledge (PMBOK), there are five key processes to developing a project schedule. PMBOK's Time Management knowledge area explains each of the inputs, tools and techniques and output in detail so you should consult the PMBOK for supplemental information. Knowing that you need to get started with developing a project schedule, let's start with the five key steps.

Step One: Define Activities

The goal of the activity definition step is to identify all the tasks required to accomplish the product. This frequently results in identifying all the work products and deliverables that comprise the project. These deliverables are found as the components of a Work Breakdown Structure (WBS). The project schedule further decomposes these deliverables into the actual activities required to complete the work.

If the project team doesn't have an established scope statement, WBS, or sufficient scope definition, you may need to host a workshop or two to gather the requirements and further develop the project schedule. Since you need to produce a project schedule by next week, you will likely create tasks in your project schedule for "Analysis" or "Scope Definition." At this point in the project, it is OK to not have all the project details. You can build activities in your project schedule to gather the information. It is perfectly acceptable to build a plan for the analysis of the project before committing to the implementation or delivery phase of the project.

Assume for now you either have a WBS available or have enough information to build a sample set of tasks to further define the scope. Once you have all the activities defined, the next step is the sequence the activities.

Step Two: Sequence Activities

At this point you've entered all the task names and have further decomposed the deliverables listed in the WBS. The next step is to sequence the activities with dependencies. During this step, you'll identify any dependencies of related tasks and document them in the project schedule. You'll need to analyze each of the tasks to understand which task has a dependency on additional tasks. In your favorite project schedule development book, be sure to read about the different types of dependency relationships include Finish-to-Start and Start-to-Start dependencies. These relationships will impact your task start and finish dates.

Step Three: Estimate Activity Resources

The next step is to identify the resources and their availability to your project. Remember that not all team members will be 100% available to your project as some team members will be working on multiple projects. In this step, you'll also assign resources to each of the tasks. I usually assign resource to tasks using the standard Gantt Chart view in Microsoft Project. For each task at the lowest point in the WBS, click on the drop down box in the Resource Names column and select the available team member.

I recommend breaking down the tasks so you can assign one task to one resource to avoid adding multiple resources to a given task. It creates a larger project schedule, but it allows me better control in allocating and tracking resources as the project executes.

Step Four: Estimate Activity Durations

With resources assigned, the next step is to estimate each task's duration. The activity's duration is the number of working periods required to complete the task. In Microsoft Project, this can be defined in days, weeks, and even months! It is also important to understand the difference of the different duration types including Fixed Work, Fixed Duration and Fixed Units. Selecting the correct duration type impacts the resource availability and the forecasted task end date.

Step Five: Develop Schedule

The next step is to analyze the project schedule and examine the sequences, durations, resources and inevitable scheduling constraints. The goal of this step is to validate the project schedule correctly models the planned work. In this step you'll not only validate the duration estimates are accurate, but validate the resource allocations are correct.

Resource leveling is a key step in ensuring the project dates are realistic and the resources are appropriately assigned. Microsoft Project has an automatic resource leveling feature, but I recommend against using it. Instead of automatic leveling, I recommend using a manual process to resolve resource over-allocation. This manual process of resource leveling is time consuming, but it results in a better end project with realistic end dates.

Once you've completed the schedule development, you'll be ready for a review with your boss for initial feedback. Once you meet with your boss to review the schedule, obtain his feedback and make the necessary changes. You'll want baseline the project schedule before you move to schedule execution. This will ensure the original dates are saved in Microsoft Project and you can compare the planned versus actual dates as the project progresses.

The next step in the PMBOK is Control the Schedule, which will be the topic of a future article. I hope these simple steps have helped you get ready for your upcoming meeting. Good luck on the project!

Scheduling terminology and techniques:

- ***Activity*** - A specific task or set of tasks that are required by the project, use up resources, and take time to complete
- ***Event*** - The result of completing one or more activities. An identifiable end state occurring at a particular time. Events use no resources.
- ***Network*** - The combination of all activities and events define the project and the activity precedence relationships
- ***Path*** - The series of connected activities (or intermediate events) between any two events in a network

- ***Critical*** - Activities, events, or paths which, if delayed, will delay the completion of the project. A project's critical path is understood to mean that sequence of critical activities that connect the project's start event to its finish event

- **An activity can be in any of these conditions:**

– It may have a successor(s) but no predecessor(s) - starts a network

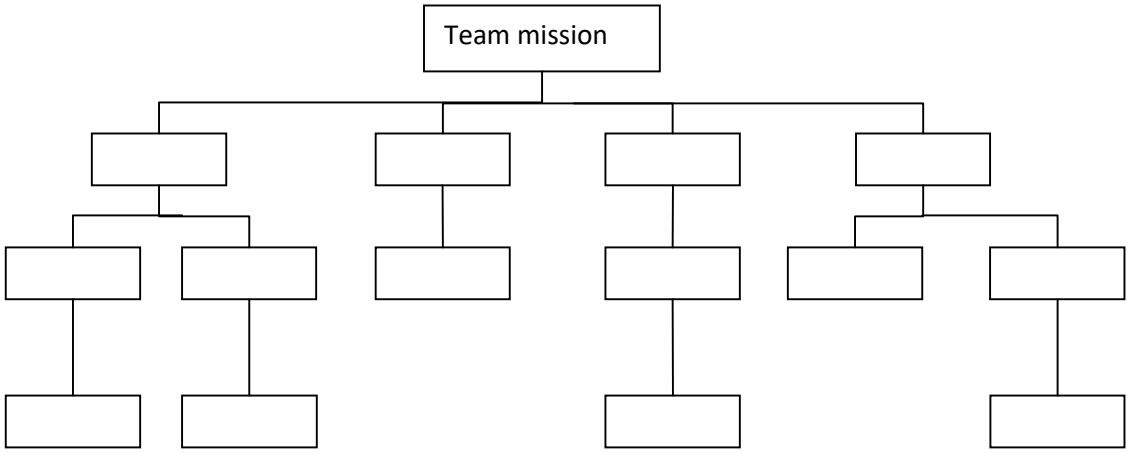
– It may have a predecessor(s) but no successor(s) - ends a network

– It may have both predecessor(s) and successor(s) - in the middle of a network

- **Interconnections from horizontal links in vertical WBS**

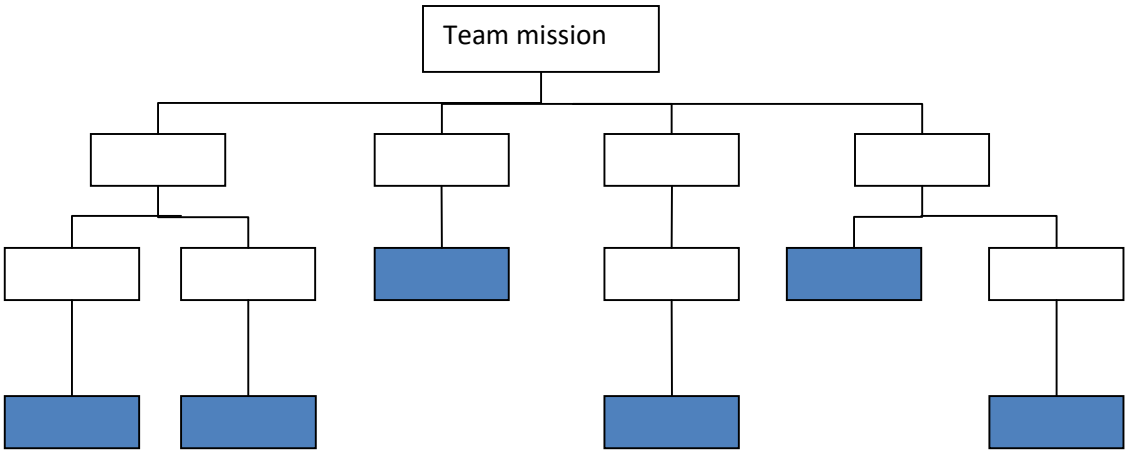
Work Breakdown Structure (WBS)

- A deliverable oriented hierarchical decomposition of the work to be executed by the project team to accomplish the team mission and create the required deliverables. The WBS defines the project scope.



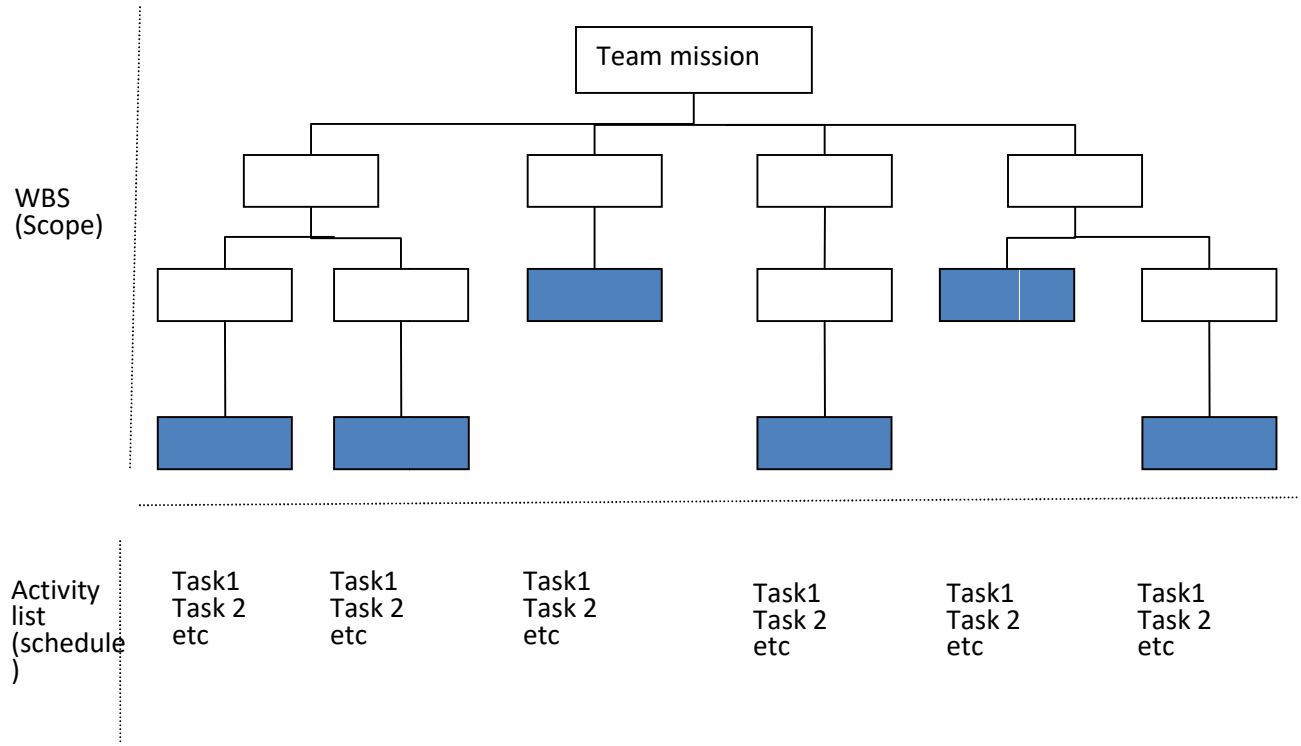
Deliverable

- Any unique and verifiable product or result that must be produced to complete a project. Usually the lowest level of the WBS.



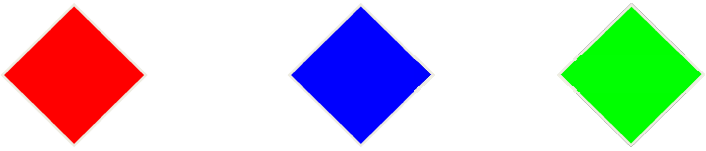
Scheduled Activity (task)

- A component of work performed to create a deliverable.



Schedule Milestone

A significant event in the project schedule, such as an event restraining future work or marking the completion of a major deliverable. A "measuring point". Milestones have "0" (zero) duration and no resource assignments



Logical Relationships

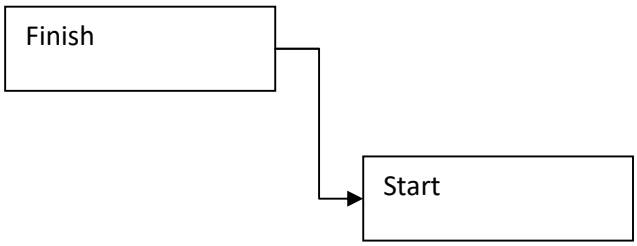
- A dependency between scheduled activities, or between a scheduled activity and a schedule milestone.

"Hard Logic" – Mandatory dependencies

"Soft Logic" – Discretionary dependencies

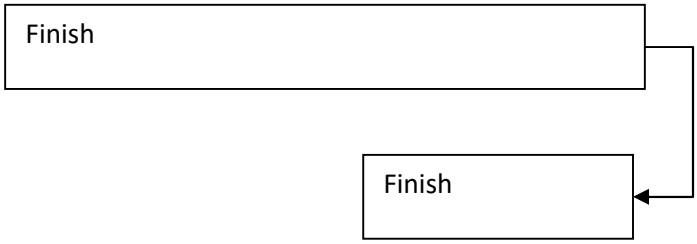
Logical Relationships

"Finish to Start" (most common)



Logical Relationships

"Finish to Finish"

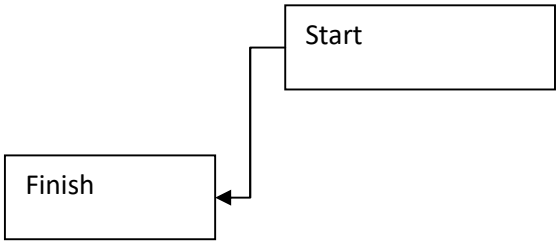


Logical Relationships

"Start to Start"

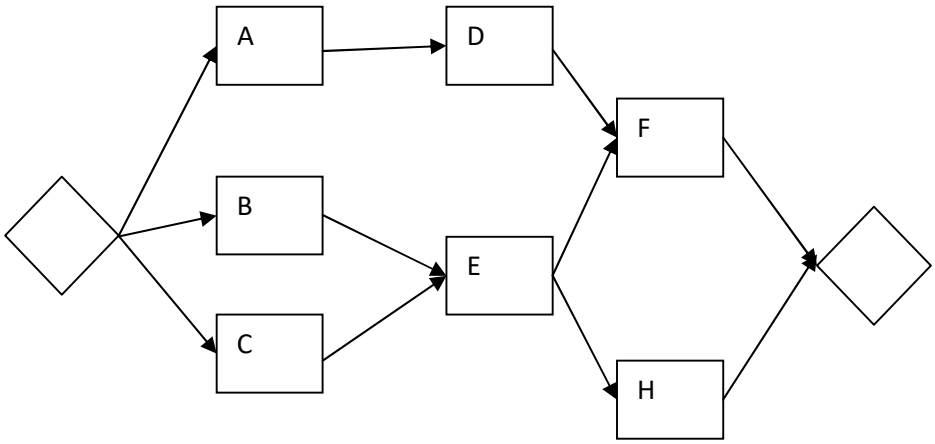
Logical Relationships

"Start to Finish" (very rare, not available in WSDOT project management software)



Precedence (Network) Diagram

A schedule diagramming technique in which schedule activities are represented by nodes and arrows shown the logical relationship between activities

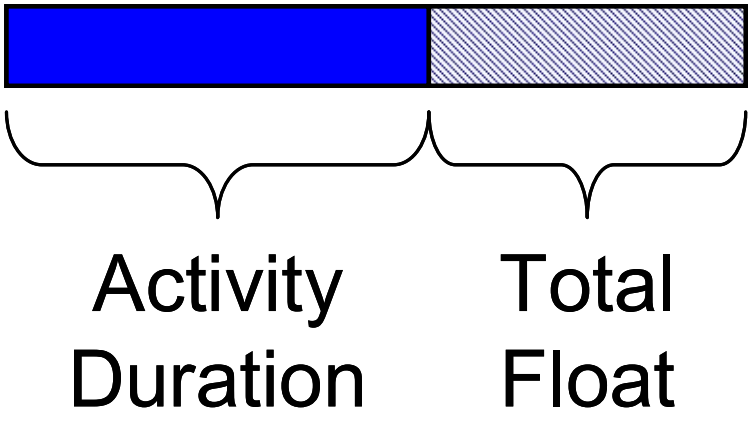


Critical Path

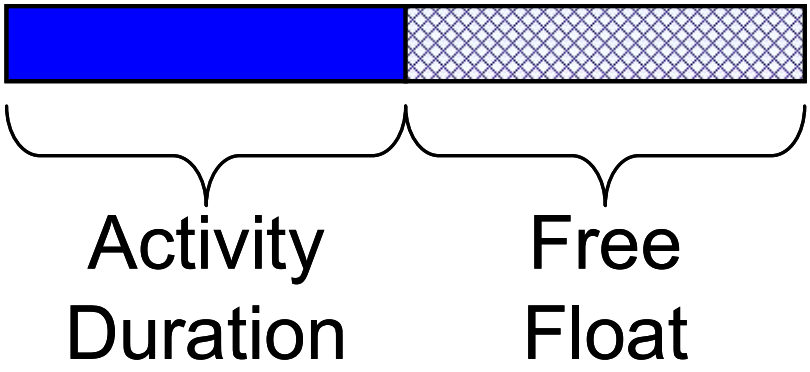
- Longest duration path through the project network
- Activities on Critical Path are called "Critical Activities"
- Critical activities have no float or slack
- A project can have more than one critical path
- Activities not on the Critical Path are called "Non Critical Activities"
- To shorten the project duration, a scheduler must modify the critical path activities or dependencies

Float (Slack)

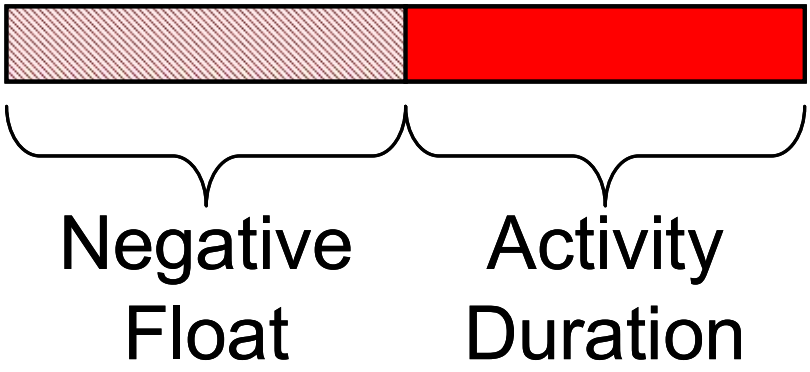
"Total Float" – The amount of time a scheduled activity has that is can be delayed or extended without affecting the project end date or violating a schedule constraint



"Free Float" – The amount of time a scheduled activity has that is can be delayed or extended without affecting the start of the next scheduled activity in the project network

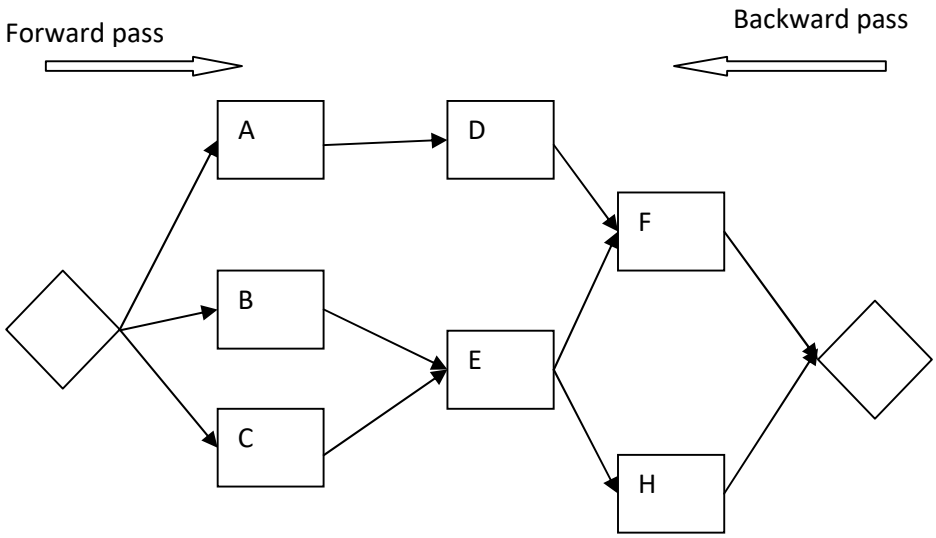


"Negative Float" – The amount of time that by which a critical activity (or milestone) misses a required date.



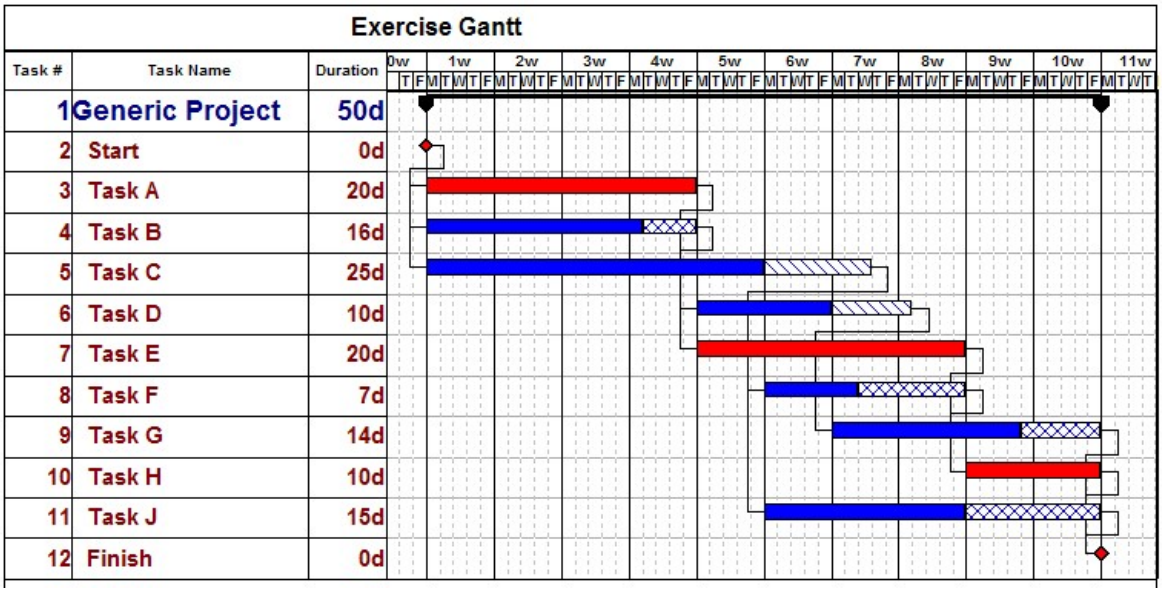
Critical Path Method (CPM)

A schedule network analysis technique used to determine the amount of schedule flexibility (float) on various network paths and to determine the minimum project duration.



Gantt chart (bar chart)

A graphic display of schedule related information. A report from the project management program.



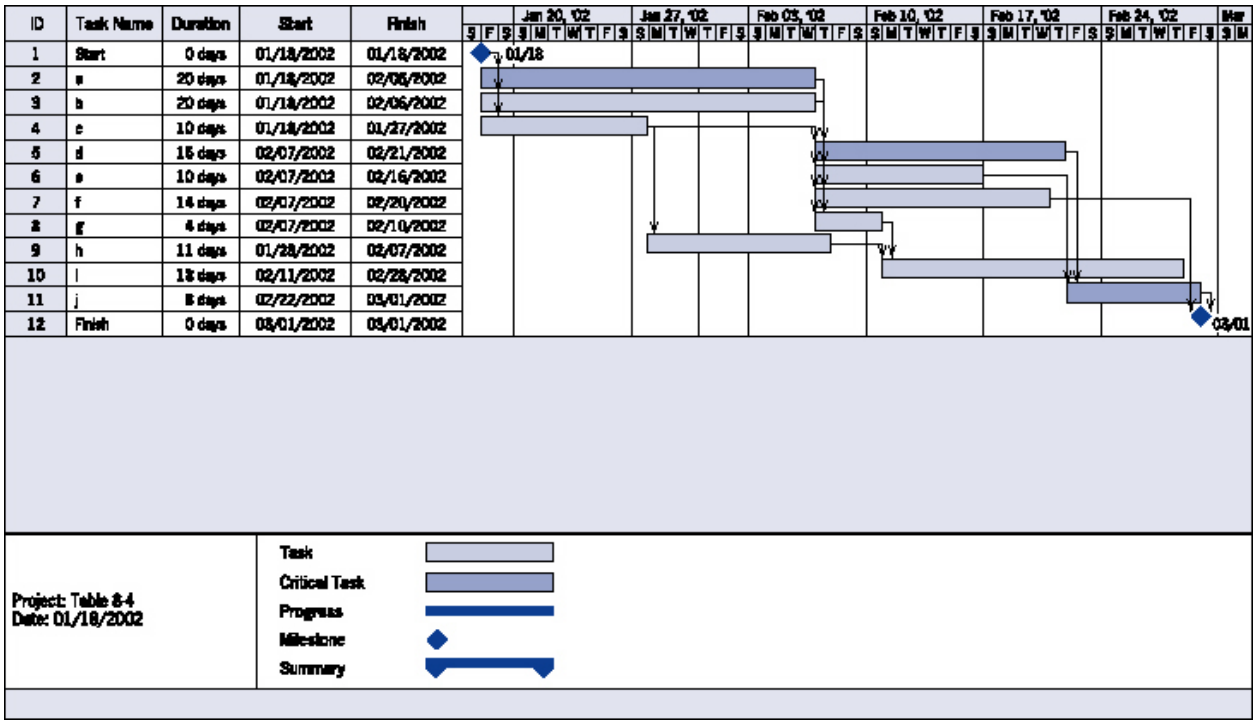
Lecture 4: Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts

Scheduling Networks

- Such networks are a powerful tool for planning and controlling a project and have the following benefits:
  - consistent framework for planning, scheduling, monitoring, and controlling the project
  - illustrates the interdependence of all tasks, work packages, and work elements
  - denotes the times when specific resources are or must be available for work on a given task
- Network benefits (cont.):
  - Facilitates communications that take place between departments and functions involved in the project.
  - determines an expected project completion date
  - identifies so-called critical activities that, if delayed, will delay the project completion time
  - identifies activities with slack that can be delayed for specific periods without penalty
- Network benefits (cont.):
  - determines the dates on which tasks may be started - or must be started if the project is to stay on schedule
  - illustrates which tasks must be coordinated to avoid resource timing conflicts
  - illustrates which tasks may run, or must be run, in parallel to achieve the predetermined project completion date
  - relieves some interpersonal conflict by clearly showing task dependencies

Gantt Charts

- Show planned and actual progress for a number of tasks displayed against a horizontal time scale
- Effective and easy-to-read method of indicating the actual current status for each set of tasks compared to planned progress for each item
- Helpful in expediting, sequencing, and reallocating resources among tasks
- Usually do not show technical dependencies



Advantage:

- easily understood / constructed / maintained
- Great visual tool to know project status (early, on time, late)
- requires frequent updating, but are easy to maintain
- Most popular way of providing a clear picture of the current state of a project (client, management, etc.)

Disadvantages:

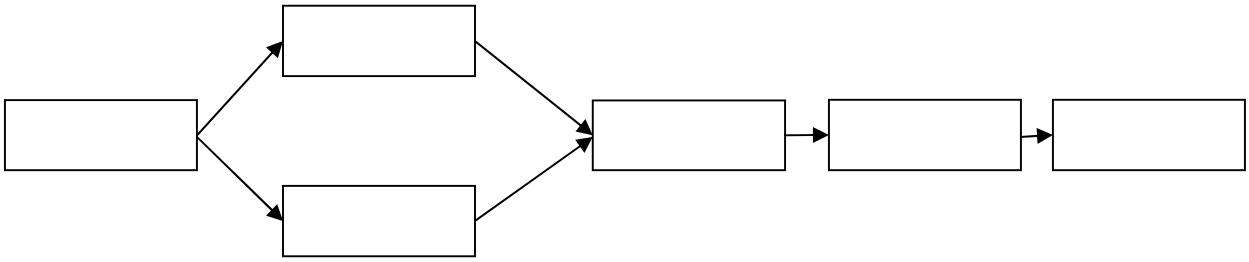
- superficial

-Hard to see activity precedents at times

PERT and CPM:

- With the exception of Gantt charts, most common approach to scheduling is the use of network techniques such as PERT and CPM
  - The Program Evaluation and Review Technique (PERT) was developed by the U.S. Navy in 1958

- The Critical Path Method (CPM) was developed by DuPont, Inc during the same time period.



Advantages

- Can visualize task relationships.
- Facilitates CP (Critical Path) Calculation
- Can see impact of decisions on downstream activities.

Disadvantages

May be complex, not easy to understand at a glance