Prof. Dr. Nizamettin AYDIN

naydin@yildiz.edu.tr
http://www.yildiz.edu.tr/~naydin

1

Project Management

The process of planning and controlling the development of system

- within a specified time frame
- at a minimum cost
- with the right functionality.

- A project manager has the primary responsibility for managing the hundreds of tasks and roles that need to be carefully coordinated.
- Nowadays, project management is an actual profession, and analysts spend years working on projects prior to tackling the management of them.
- There also is a variety of project management software available like Microsoft Project, Plan View, and PMOffice that support project management activities.
- Although training and software are available to help project managers, unreasonable demands set by project sponsors and business managers can make project management very difficult.

3

- A critical success factor for project management is to start with a realistic assessment of the work that needs to be accomplished and then manage the project according to that assessment.
- This can be achieved by carefully following the four steps:
  - identifying the project size,
  - creating and managing the workplan,
  - staffing the project,
  - coordinating project activities.
- The project manager ultimately creates
  - a workplan,
  - staffing plan,
  - standards list,

  which are used and refined throughout the entire SDLC.

4

IDENTIFYING PROJECT SIZE

- The science of project management is in making trade-offs among three important concepts:
  - the size of the system
    - in terms of what it does
  - the time to complete the project
    - when the project will be finished
  - the cost of the project.
- Think of these three things as interdependent levers that the project manager controls throughout the SDLC.

5

IDENTIFYING PROJECT SIZE

- Whenever one lever is pulled, the other two levers are affected in some way.
- For example, if a project manager needs to readjust a deadline to an earlier date, then the only solution is
  - to decrease the size of the system
    - by eliminating some of its functions
  - or to increase costs
    - by adding more people or having them work overtime.
- Often, a project manager will have to work with the project sponsor to change the goals of the project,
  - such as developing a system with less functionality or extending the deadline for the final system,

  so that the project has reasonable goals that can be met.

6

IDENTIFYING PROJECT SIZE

- Therefore, in the beginning of the project, the manager needs to estimate each of these levers and then continuously assess how to roll out the project in a way that meets the organization's needs.
- Estimation is the process of assigning projected values for time and effort, and it can be performed manually or with the help of an estimation software package.
- The estimates developed at the start of a project are usually based on a range of possible values
  - e.g., the design phase will take three to four months
- and gradually become more specific as the project moves forward
  - e.g., the design phase will be completed on March 22.

7

IDENTIFYING PROJECT SIZE

- The numbers used to calculate these estimates can come from several sources. They can be
  - provided with the methodology that is used,
  - taken from projects with similar tasks and technologies,
  - provided by experienced developers.
- One of the greatest strengths of systems consulting firms is the past experience that they offer to a project;
  - they have estimates and methodologies that have been developed and honed over time and applied to hundreds of projects.

8

Estimating the time required to build a system

- The simplest method uses the amount of time spent in the planning phase to predict the time required for the entire project.
- The idea is that
  - a simple project will require little planning
  - a complex project will require more planning,
- so using the amount of time spent in the planning phase is a reasonable way to estimate overall project time requirements.
- With this approach, you take the time spent in the planning phase and use industry standard percentages to calculate estimates for the other SDLC phases

9

- Industry standards suggest that a typical business application system spends
  - 15 percent of its effort in the planning phase,
  - 20 percent in the analysis phase,
  - 35 percent in the design phase,
  - 30 percent in the implementation phase.
- This would suggest that if a project takes four months in the planning phase, then the rest of the project likely will take a total of 22.66 person-months (4 ÷ 0.15 = 22.66).
- These same industry percentages are then used to estimate the amount of time in each phase.
- The obvious limitation of this approach is that it can be difficult to take into account the specifics of your individual project, which may be simpler or more difficult than the typical project.

10

- Estimating Project Time Using the Planning Phase Approach

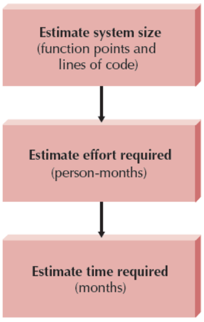
	Planning	Analysis	Design	Implementation
Typical industry standards for business applications	15%	20%	35%	30%
Estimates based on actual figures for first stages of SDLC	Actual: 4 person-months	Estimated: 5.33 person-months	Estimated: 9.33 person-months	Estimated: 8 person-months

11

- The second approach to estimation is the function point approach,
- uses a three-step process:
  - First, the project manager estimates the size of the project in terms of the number of lines of code the new system will require.
  - This size estimate is then converted into the amount of effort required to develop the system in terms of the number of person-months.
  - The estimated effort is then converted into an estimated schedule time in terms of the number of months from start to finish.

12

- Estimating Project Time Using the Function Point Approach



Step 1: Estimate System Size

- The size of a project is estimated using function points
  - a concept developed in 1979 by Allen Albrecht of IBM.
- A function point is a measure of program size that is based on the system's number and complexity of
  - inputs,
  - outputs,
  - queries,
  - files,
  - program interfaces.

Function Point-Estimation Worksheet

- To calculate the function points for a project, components are listed on a worksheet to represent the major elements of the system.

System Components:

Description	Total Number	Complexity: Low	Complexity: Medium	Complexity: High	Total
Inputs	6	3 × 3	2 × 4	1 × 6	23
Outputs	19	4 × 4	10 × 5	5 × 7	101
Queries	10	7 × 3	0 × 4	3 × 6	39
Files	15	0 × 7	15 × 10	0 × 15	150
Program Interfaces	3	1 × 5	0 × 7	2 × 10	25
Total Unadjusted Function Points (TUFP):					338

Function Point-Estimation Worksheet

- For example, data-entry screens are kinds of inputs, reports are outputs, and database queries are kinds of queries.
- The project manager records the total number of each component that the system will include, and then (s)he breaks down the number to show the number of components that have low, medium, and high complexity.
- In the figure, there are nineteen outputs that need to be developed for the system, four of which have low complexity, ten that have medium complexity, and five that are very complex.
- After each line is filled in, a total number of points are calculated per line by multiplying each number by a complexity index.
- The line totals are added up to determine the total unadjusted function points (TUFP) for the project.

Function Point-Estimation Worksheet

- The complexity of the overall system is greater than the sum of its parts.
- Things like the familiarity of the project team with the business area and the technology that will be used to implement the project also may influence how complex a project will be.
  - A project that is very complex for a team with little experience might have little complexity for a team with lots of experience.
- To create a more realistic size for the project, a number of additional system factors, such as
  - end-user efficiency, reusability, data communications

  are assessed in terms of their effect on the project's complexity.

Data communications	3
Heavy use configuration	0
Transaction rate	0
End-user efficiency	0
Complex processing	0
Installation ease	0
Multiple sites	0
Performance	0
Distributed functions	2
Online data entry	2
Online update	0
Reusability	0
Operational ease	0
Extensibility	0
Total Processing Complexity (PC):	7

Function Point-Estimation Worksheet

Overall System:

0 = no effect on processing complexity

3 = great effect on processing complexity

- Adjusted Processing Complexity (APC):

  0.65 + (0.01 x 7) = 0.72

- Total Adjusted Function Points (TAFP):

  0.72 (APC) x 338 (TUFP) = 243

Data communications	3
Heavy use configuration	0
Transaction rate	0
End-user efficiency	0
Complex processing	0
Installation ease	0
Multiple sites	0
Performance	0
Distributed functions	2
Online data entry	2
Online update	0
Reusability	0
Operational ease	0
Extensibility	0
Total Processing Complexity (PC):	7

Function Point-Estimation Worksheet

- These assessments are totaled and placed into a formula to calculate an adjusted project complexity (APC) score.
- The TUFP value is multiplied by the APC value to determine the ultimate size of the project in terms of total adjusted function points(TAFP).
- This number should give the project manager a reasonable idea as to how big the project will be.
- Sometimes a shortcut is used to determine the complexity of the project.
  - Instead of calculating the complexity for the 14 factors listed in the Figure, project managers choose to assign an APC value that ranges from 0.65 for very simple systems to 1.00 for "normal" systems to as much as 1.35 for complex systems and multiply the value to the TUFP score.

- In the Planning Phase, the exact nature of the system has not yet been determined,
  - so it is impossible to know exactly how many inputs, outputs, and so forth will be in the system.
  - It is up to the project manager to make an intelligent guess.
- Some people feel that using function points early on in a project is not practical for this reason.
- However, It is likely that function points can be a useful tool for understanding a project's size at any point in the SDLC.

- Later in the project, once more is known about the system,
  - the project manager will revise the estimates using this better knowledge to produce more accurate results.
- Once you have estimated the number of function points, you need to convert the number of function points into the lines of code that will be required to build the system.
- The number of lines of code depends on the programming language you choose to use.
  - Following figure presents a very rough conversion guide for some popular languages.

Converting from Function Points to Lines of Code

Language	Approximate Number of Lines of Code per Function Point
C	130
COBOL	110
Java	55
C++	50
Turbo Pascal	50
Visual Basic	30
PowerBuilder	15
HTML	15
Packages (e.g., Access, Excel)	10–40

Source: Capers Jones, Software Productivity Research, http:/www.spr.com

- For example, the system discussed earlier has 243 function points.
- If you were to develop the system in COBOL, it would typically require approximately 26,730 lines of code to write it.
- Conversely, if you were to use Visual Basic, it typically would take 7,290 lines of code.
- If you could develop the system using a package such as Excel or Access, it would take between 2,430 and 9,720 lines of code.
- There is a great range for packages, because different packages enable you to do different things, and not all systems can be built using certain packages.

- Sometimes you end up writing lots of extra code to do some simple function because the package does not have the capabilities you need.
- There is also a very important message from the data in the figure in slide 22.
- Since there is a direct relationship between lines of code and the amount of effort and time required to develop a system,
  - the choice of development language has a significant impact on the time and cost of projects.

Step 2: Estimate Effort Required

15Kasim2k11

- Once an understanding is reached about the size of the system, the next step is
  - to estimate the effort that is required to build it.
- Effort is a function of the system size combined with production rates
  - how much work someone can complete in a given time.
- Much research has been done on software production rates.
  - One of the most popular algorithms, the COCOMO (COnstructive COst MOdel), was designed by Barry W. Boehm to convert a lines-of-code estimate into a person-month estimate.
  - There are different versions of the COCOMO model that vary based on the complexity of the software, the size of the system, the experience of the developers, and the type of software you are developing.

25

Step 2: Estimate Effort Required

- For small to moderate-size business software projects (i.e., 100,000 lines of code and ten or fewer programmers), the model is quite simple:

  effort (in person-months) = 1.4 × thousands of lines of code
- For example,
  - let's suppose that we were going to develop a business software system requiring 10,000 lines of code.
- This project would typically take 14 person-months to complete.
- If the system in figure in slides 15-17 were developed in COBOL (which equates to 26,730 lines of code),
  - it would require about 37.42 person-months of effort.

26

COCOMOs

- http://cost.jsc.nasa.gov/COCOMO.html#COCOMO
  - This model is based on Barry Boehm's COCOMO. This is the top-level model, Basic COCOMO, which is applicable to the large majority of software projects.
  - However, this model is obsolete and should only be used for demonstration purposes.
  - Following model (COCOMO II model at USC (University of Southern California)) should be used for real estimates.
- http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html

27

COCOMOs

- COCOPRO implements Boehm's COCOMO technique for estimating costs of software projects. It supports the intermediate COCOMO model, and allows automatic calibration of the model to a cost history database.
- COOLSoft uses a hybrid of intermediate and detailed versions of COCOMO. This allows for the reuse of existing code, development of new code, the purchase and integration of third party code, and hardware integration.
- Costar is a software cost estimation tool based on COCOMO. A software project manager can use Costar to produce estimates of a project's duration, staffing levels, effort, and cost. Costar is an interactive tool that permits managers to make trade-offs and experiment with what-if analyses to arrive at the optimal project plan.
- Revised Intermediate COCOMO (REVIC) is available for downloading from the US Air Force Cost Analysis Agency (AFCAA).
- TAMU COCOMO is an on-line version of COCOMO from Texas A&M University.

28

Step 3: Estimate Time Required

- Once the effort is understood, the optimal schedule for the project can be estimated.
- Historical data or estimation software can be used as aids,

  or one rule of thumb is to determine schedule using the following equation:
  - schedule time (months) = 3.0 × person-months$^{1/3}$
- This equation is widely used, although the specific numbers vary
  - e.g., some estimators may use 3.5 or 2.5 instead of 3.0.

29

Step 3: Estimate Time Required

- The equation suggests that a project that has an effort of 14 person-months should be scheduled to take a little more than 7 months to complete.
- Considering the figure in slides 15-17 example,
  - the 37.42 person-months would require a little over 10 months.
- It is important to note that this estimate is for the analysis, design, and implementation phases;
  - it does not include the planning phase.

30

Creating and Managing the Workplan

Once a project manager has a general idea of the size and approximate schedule for the project,

(s)he creates a workplan,

which is a dynamic schedule that records and keeps track of all of the tasks that need to be accomplished over the course of the project.

- The workplan lists each task, along with important information about it, such as
  - when it needs to be completed,
  - the person assigned to do the work,
  - any deliverables that will result
- The level of detail and the amount of information captured by the workplan depend on the needs of the project
  - the detail usually increases as the project progresses.
- Usually, the workplan is the main component of the project management software.

Workplan Information	Example
Name of the task	Perform economic feasibility
Start date	Jan 05, 2003
Completion date	Jan 19, 2003
Person assigned to the task	Project sponsor; Mary Smith
Deliverable(s)	Cost–benefit analysis
Completion status	Open
Priority	High
Resources that are needed	Spreadsheet software
Estimated time	16 hours
Actual time	14.5 hours

- To create a workplan, the project manager
  - identifies the tasks that need to be accomplished
  - determines how long they will take.
- Then the tasks are organized within a workplan and presented graphically using
  - Gantt chart.
  - PERT chart.
- All of these techniques help a project manager understand and manage the project's progress over time.

Identify Tasks

- The overall objectives for the system should be listed on the system request,
  - Identifying all of the tasks that need to be accomplished to meet those objectives is the project manager's job .
- One approach for identifying tasks is
  - to get a list of tasks that has already been developed and to modify it.
- There are standard lists of tasks, or methodologies, that are available for use as a starting point.
  - A methodology is a formalized approach to implementing the SDLC
    - i.e., it is a list of steps and deliverables.

- A project manager can take an existing methodology, select the steps and deliverables that apply to the current project, and add them to the workplan.
- If an existing methodology is not available within the organization, methodologies can be purchased from consultants or vendors, or books can serve as guidance.
- Using an existing methodology is the most popular way to create a workplan because most organizations have a methodology that they use for projects.

- If a project manager prefers to begin from scratch, (s)he can use a structured, topdown approach whereby high-level tasks are first defined and then these are broken down into subtasks.
  - For example, next figure shows a list of high-level tasks that are needed to implement a new IT training class.
  - Some of the main steps in the process include identifying vendors, creating and administering a survey, and building new classrooms.
  - Each step is then broken down in turn and numbered in a hierarchical fashion.
  - There are eight subtasks for creating and administering a survey, and there are three subtasks that comprise the review initial survey task.
  - A list of tasks hierarchically numbered in this way is called a work breakdown structure (WBS), and it is the backbone of the project workplan.

Task Number	Task Name	Duration (in weeks)	Dependency	Status
1	Identify vendors	2		Complete
2	Review training materials	6	1	Complete
3	Compare vendors	2	2	In Progress
4	Negotiate with vendors	3	3	Open
5	Develop communications information	4	1	In Progress
6	Disseminate information	2	5	Open
7	Create and administer survey	4	6	Open
7.1	Create initial survey	1		Open
7.2	Review initial survey	1	7.1	Open
7.2.1	Review by Director of IT Training	1		Open
7.2.2	Review by Project Sponsor	1		Open
7.2.3	Review by Representative Trainee	1		Open
7.3	Pilot test initial survey	1	7.1	Open
7.4	Incorporate survey changes	1	7.2, 7.3	Open
7.5	Create distribution list	0.5		Open
7.6	Send survey to distribution list	0.5	7.4, 7.5	Open
7.7	Send follow-up message	0.5	7.6	Open
7.8	Collect completed surveys	1	7.6	Open
8	Analyze results and choose vendor	2	4, 7	Open
9	Build new classrooms	11	1	In Progress
10	Develop course options	3	8, 9	Open

37

- The number of tasks and level of detail depend on the complexity and size of the project.
- The larger the project, the more important it becomes to define tasks at a low level of detail so that essential steps are not overlooked.

38

- There are two basic approaches to organizing a work breakdown structure:
  - by SDLC phase
  - by product.
- For example, if a firm decided that it needed to develop a Web site, the firm could create a work breakdown structure based on the SDLC phases:
  - planning, analysis, design, and implementation.
  - In this case, a typical task that would take place during planning would be feasibility analysis.
  - This task would be broken down into the different types of feasibility analysis:
    - technical, economic, and organizational.
    - Each of these would be further broken down into a set of subtasks.

39

- Alternatively, the firm could organize the workplan along the lines of the different products to be developed.
- For example, in the case of a Web site, the products could include
  - applets, application servers, database servers, the various sets of Web pages to be designed, a site map, and so on.
- Then these would be further decomposed into the different tasks associated with the phases of the SDLC.
- Previous figure depicts the tasks necessary for creating a new IT training class.
- Either way, once the overall structure is determined, tasks are identified and included in the work breakdown structure of the Workplan.

40

The Project Workplan

- the mechanism that is used to manage the tasks that are listed in the work breakdown structure.
- It is the project manager's primary tool for managing the project.
  - the project manager can tell
    - if the project is ahead or behind schedule,
    - how well the project was estimated,
    - what changes need to be made to meet the project deadline.
- Basically, the workplan is a table that lists all of the tasks in the work breakdown structure along with important task information, such as
  - the people who are assigned to perform the tasks,
  - the actual hours that the tasks took, and
  - the variances between estimated and actual completion times

41

The Project Workplan

- At a minimum, the information should include
  - the duration of the task,
  - the current statuses of the tasks (i.e., open, complete),
  - the task dependencies,
    - which occur when one task cannot be performed until another task is completed.
- Key milestones, or important dates, are also identified on the workplan. Important milestones could be
  - presentations to the approval committee,
  - the start of end-user training, a company retreat,
  - the due date of the system prototype

42

Gantt Chart

- A horizontal bar chart
  - shows the same task information as the project workplan, but in a graphical way.
- Sometimes a picture really is worth a thousand words,
  - the Gantt chart can communicate the high-level status of a project much faster and easier than the workplan.
- Creating a Gantt chart is simple and can be done using
  - a spreadsheet package,
  - graphics software (e.g., Microsoft VISIO),
  - a project management package.

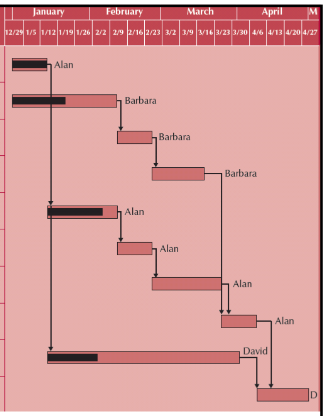
43

Gantt Chart

- Tasks are listed as rows in the chart,
- Time is listed across the top in increments based on the needs of the projects.
  - A short project may be divided into hours or days;
  - A medium-sized project may be represented using weeks or months.
- Horizontal bars are drawn to represent the duration of each task;
  - the bar's beginning and end mark exactly when the task will begin and end.
    - As people work on tasks, the appropriate bars are filled in proportionately to how much of the task is finished.
- Too many tasks on a Gantt chart can become confusing,
  - it's best to limit the number of tasks to around twenty to thirty.
    - If there are more tasks, break them down into subtasks and create Gantt charts for each level of detail.

44

ID	Task Name	Duration	Start	Finish	Prede
1	Identify vendors	2 wks	Wed 1/1/03	Tue 1/14/03	
2	Review training materials	6 wks	Wed 1/1/03	Tue 2/11/03	
3	Compare vendors	2 wks	Wed 2/12/03	Tue 2/25/03	2
4	Negotiate with vendors	3 wks	Wed 2/26/03	Tue 3/18/03	3
5	Develop communications information	4 wks	Wed 1/15/03	Tue 2/11/03	1
6	Disseminate information	2 wks	Wed 2/12/03	Tue 2/25/03	5
7	Create and administer survey	4 wks	Wed 2/26/03	Tue 3/25/03	6
8	Analyze results and choose	2 wks	Wed 3/26/03	Tue 4/8/03	4, 7
9	Build new classroom	11 wks	Wed 1/15/03	Tue 4/1/03	1
10	Develop course options	3 wks	Wed 4/9/03	Tue 4/29/03	8, 9



45

Gantt Chart

- There are many things a project manager can see by looking quickly at a Gantt chart.
  - How long tasks are and how far along they are
  - Which tasks are sequential, which tasks occur at the same time, and which tasks overlap in some way.
- There are a few special notations that can be placed on a Gantt chart.
  - Project milestones are shown using upside-down triangles or diamonds.
  - Arrows are drawn between the task bars to show task dependencies.
  - Sometimes, the names of people assigned to each task are listed next to the task bars to show what human resources have been allocated to each task.

46

PERT Chart

- Program Evaluation and Review Technique lays out the project tasks in a flowchart.
- a network analysis technique
  - can be used when the individual task time estimates are fairly uncertain.
- Instead of simply putting a point estimate for the duration estimate, PERT uses three time estimates:
  - optimistic,
  - most likely,
  - pessimistic.

47

PERT Chart

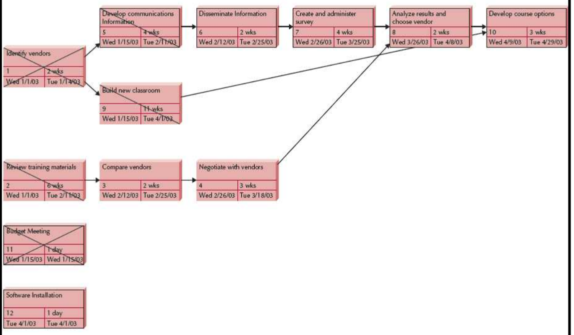
- It then combines the three estimates into a single weighted average estimate using the following formula:

PERT weighted average =

$$\frac{\text{optimistic estimate} + (4 \times \text{most likely estimate}) + \text{pessimistic estimate}}{6}$$

- The PERT chart is drawn as a node and arc type of graph that shows the time estimates in the nodes and the task dependencies on the arcs.
- Each node represents an individual task, and a line connecting two nodes represents the dependency between two tasks.
- Usually partially completed tasks are displayed with a diagonal line through the node, and completed tasks contain crossed lines.

48



49

PERT Chart

- the best way to communicate task dependencies
  - because they lay out the tasks in the order in which they need to be completed.
- The Critical Path Method (CPM) simply allows the identification of the critical path in the network.
  - The critical path is the longest path from the project inception to completion.
  - The critical path shows all of the tasks that must be completed on schedule for a project as a whole to finish on schedule.
- If any of the tasks on the critical path takes longer than expected, the entire project will fall behind.

50

PERT Chart

- Each task on the critical path is a critical task.
  - CPM can be used with or without PERT.
- The benefit of using project management software packages like Microsoft Project is that the workplan can be input once, and then the software can display the information in many different formats.
- You can toggle between the workplan, a Gantt chart, and a PERT chart depending on your project management needs.

51

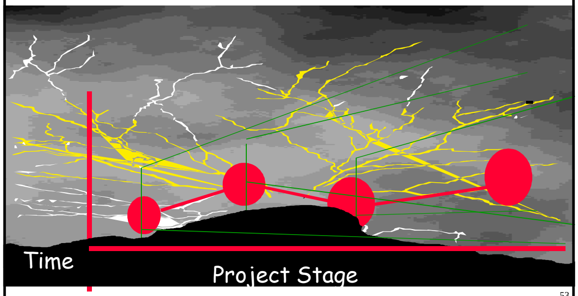
Refining Estimates

- The estimates that are produced during the planning phase will need to be refined as the project progresses.
- This does not mean that estimates were poorly done at the start of the project,
  - it is virtually impossible to develop an exact assessment of the project's schedule before the analysis and design phases are conducted.
- A project manager should expect to be satisfied with broad ranges of estimates
  - that become more and more specific as the project's product becomes better defined.

52

The Hurricane Model

- Estimating what an IS development project will cost, how long it will take, and what the final system will actually do follows a hurricane model.



53

The Hurricane Model

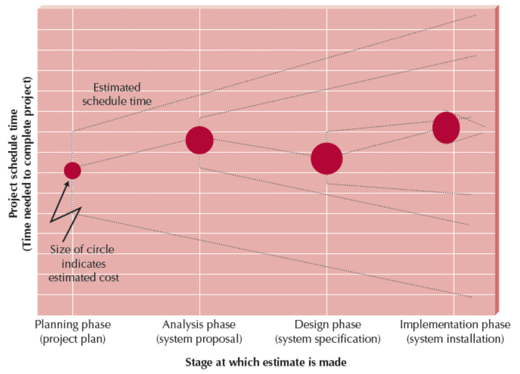
- When storms and hurricanes first appear in the Atlantic or Pacific, forecasters
  - watch their behavior
  - attempt to predict when and where the storms will hit and what damage they will do when they arrive.
- As storms move closer to North America, forecasters
  - refine their tracks
  - develop better predictions about where and when they are most likely to hit and their force when they do.
- The predictions become more and more accurate as the storms approach a coast, until they finally arrive.

54

The Hurricane Model

- In the Planning Phase when a system is first requested,
  - the project sponsor and project manager attempt to predict
    - how long the SDLC will take,
    - how much it will cost,
    - what it will ultimately do when it is delivered (i.e., its functionality).
  - However, the estimates are based on very little knowledge of the system.
- As the system moves into the Analysis Phase,
  - more information is gathered,
  - the system concept is developed,
  - the estimates become even more accurate and precise.
- As the system moves closer to completion,
  - the accuracy and precision increase until the final system is delivered

The Hurricane Model



Margins of Error in Cost and Time Estimates

- According to one of the leading experts in software development,
  - a well-done project plan (prepared at the end of the planning phase) has a 100 percent margin of error for project cost
  - 25 percent margin of error for schedule time.
- In other words, if a carefully done project plan estimates that a project will cost $100,000 and take twenty weeks,
  - the project will actually cost between $0 and $200,000
  - take between fifteen and twenty-five weeks.

		Typical Margins of Error for Well-Done Estimates	
Phase	**Deliverable**	**Cost (%)**	**Schedule Time (%)**
Planning phase	System request	400	60
	Project plan	100	25
Analysis phase	System proposal	50	15
Design phase	System specifications	25	10

Source: Barry W. Boehm and colleagues, "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," in J. D. Arthur and S. M. Henry (editors) *Annals of Software Engineering Special Volume on Software Process and Product Measurement*, Amsterdam: J. C. Baltzer AG Science Publishers, 1995.

- It is important to note that these margins of error apply only to well-done plans; a plan developed without much care has a much greater margin of error.

- What happens if you overshoot an estimate ?
  - e.g., the Analysis Phase ends up lasting two weeks longer than expected?
- There are number of ways to adjust future estimates.
- If the project team finishes a step ahead of schedule,
  - most project managers shift the deadlines sooner by the same amount
  - but do not adjust the promised completion date.
- The challenge, however, occurs when the project team is late in meeting a scheduled date.
- Next slide presents three possible responses to missed schedule dates.

Assumptions	Actions	Level of Risk
If you assume the rest of the project is simpler than the part that was late and is also simpler than believed when the original schedule estimates were made, you can make up lost time	Do not change schedule.	High risk
If you assume the rest of the project is simpler than the part that was late and is no more complex than the original estimate assumed, you can't make up the lost time, but you will not lose time on the rest of the project.	Increase the entire schedule by the total amount of time that you are behind (e.g., if you missed the scheduled date by two weeks, move the rest of the schedule dates to two weeks later). If you included padded time at the end of the project in the original schedule, you may not have to change the promised system delivery date; you'll just use up the padded time.	Moderate risk
If you assume that the rest of the project is as complex as the part that was late (your original estimates too optimistic), then all the scheduled dates in the future underestimate the real time required by the same percentage as the part that was late.	Increase the entire schedule by the percentage of weeks that you are behind (e.g., if you are two weeks late on part of the project that was supposed to take eight weeks, you need to increase all remaining time estimates by 25 percent). If this moves the new delivery date beyond what is acceptable to the project sponsor, the scope of the project must be reduced.	Low risk

- It is recommend that if an estimate proves too optimistic early in the project,
  - do not expect to make up for lost time
    - very few projects end up doing this.
- Instead, change your future estimates to include an increase similar to the one that was experienced.
- For example, if the first phase was completed 10 percent over schedule,
  - increase the rest of your estimates by 10 percent.

61

Scope Management

- It is assumed that a project will be safe from scheduling problems because it is carefully estimated and planned.
- However, schedule and cost overruns occurs after the project is underway as a result of scope creep.
- Scope creep happens when new requirements are added to the project after the original project scope was defined and frozen.
- It can happen for many reasons:
  - users may suddenly understand the potential of the new system and realize new functionality that would be useful;
  - developers may discover interesting capabilities to which they become very attached;
  - a senior manager may decide to let this system support a new strategy that was developed at a recent board meeting.

62

Scope Management

- After the project begins, it becomes increasingly difficult to address changing requirements.
- Therefore, the project manager plays a critical role in managing this change to keep scope creep to a minimum.
- The keys are to identify the requirements as well as possible in the beginning of the project and to apply analysis techniques effectively.
  - For example, if needs are fuzzy at the project's onset, a combination of intensive meetings with the users and prototyping could be used so that users experience the requirements and better visualize how the system could support their needs.
- In fact, the use of meetings and prototyping has been found to reduce scope creep to less than 5 percent on a typical project.

63

Scope Management

- Sometimes changes cannot be incorporated into the present system even though they truly would be beneficial.
- In this case, these additions to scope should be recorded as future enhancements to the system.
- The project manager can offer to provide functionality in future releases of the system, thus getting around telling someone no.

64

Managing Scope

- Scope creep -- a major cause of development problems
  - JAD and prototyping
  - Formal change approval
  - Charging for changes

65

Timeboxing

- So far, projects that are task oriented have been described.
  - projects that have a schedule that is driven by the tasks that need to be accomplished,
    - so the greater number of tasks and requirements, the longer the project will take.
- Some companies have little patience for development projects that take a long time,
- They take a time-oriented approach that places meeting a deadline above delivering functionality.

66

Timeboxing

- In many software applications;
  - most users rely on only a small subset of their capabilities.
- In developing a system;
  - typically 75 percent of a system can be provided relatively quickly,
  - the remaining 25 percent of the functionality demanding most of the time.

67

Timeboxing

- To resolve this incongruency,
  - timeboxing is employed,
    - especially when using rapid application development (RAD) methodologies.
- This technique sets a fixed deadline for a project and delivers the system by that deadline no matter what, even if functionality needs to be reduced.
- Timeboxing ensures that project teams don't get hung up on the final finishing touches that can drag out indefinitely.
- It satisfies the business by providing a product within a relatively fast time frame.

68

Timeboxing

- There are several steps to implement timeboxing on a project.
- First, set the date of delivery for the proposed goals.
  - The deadline should not be impossible to meet,
    - so it is best to let the project team determine a realistic due date.
- Next, build the core of the system to be delivered;
  - you will find that timeboxing helps create a sense of urgency and helps keep the focus on the most important features.
- Because the schedule is absolutely fixed,
  - functionality that cannot be completed needs to be postponed.
    - It helps if the team prioritizes a list of features beforehand to keep track of what functionality the users absolutely need.

69

Timeboxing

- Quality cannot be compromised, regardless of other constraints,
  - so it is important that the time allocated to activities is not shortened unless the requirements are changed
    - e.g., don't reduce the time allocated to testing without reducing features.
- At the end of the time period, a highquality system is delivered;
  - likely, future iterations will be needed to make changes and enhancements, and the timeboxing approach can be used once again.

70

Timeboxing

- Fixed deadline
  - Make deadline possible
  - Set by development group
- Reduced functionality, if necessary
- Fewer "finishing touches"

71

Steps for Timeboxing

1. Set the date for system delivery.
2. Prioritize the functionality that needs to be included in the system.
3. Build the core of the system (the functionality ranked as most important).
4. Postpone functionality that cannot be provided within the time frame.
5. Deliver the system with core functionality.
6. Repeat steps 3 through 5, to add refinements and enhancements.

72

Evolutionary Work Breakdown Structures and Iterative Workplans

- Most approaches to developing conventional WBSs tend to have three underlying problems:

1. They tend to be focused on the design of the information system being developed.
   - As such, the creation of the WBS forces the decomposition of the system design and the tasks associated with creating the design of the system prematurely.
   - Where the problem domain is well understood, tying the structure of the workplan to the product to be created makes sense.
   - However, in cases where the problem domain is not well understood, the analyst must commit to the architecture of the system being developed before the requirements of the system are fully understood.

Evolutionary Work Breakdown Structures and Iterative Workplans

2. They tend to force too many levels of detail very early on in the SDLC for large projects or they tend to allow too few levels of detail for small projects.
   - Since the primary purposes of a WBS is to allow cost estimation and scheduling to take place, in conventional approaches to planning, the WBS must be done "correctly and completely" at the beginning of the SDLC.
     - To say the least, this is a very difficult task to accomplish with any degree of validity.
   - In such cases, it is no wonder why cost and schedule estimation for many information systems development projects tend to be wildly inaccurate.

Evolutionary Work Breakdown Structures and Iterative Workplans

3. Since they are project specific, they are very difficult to compare across projects.
   - This leads to ineffective learning across the organization.
   - Without some standard approach to create WBSs, it is difficult for project managers to learn from previous projects managed by others.
   - This tends to encourage the "reinventing of wheels" and allows managers to make the same mistakes that previous managers have made.

- Evolutionary WBSs allow the analyst to address all three of these problems by allowing the development of an iterative workplan.

Evolutionary Work Breakdown Structures and Iterative Workplans

- 1st, they are organized in a standard manner across all projects: by
  - workflows,
  - phases,
  - tasks.
- This decouples the structure of an evolutionary WBS from the structure of the design of the product.
- This prevents prematurely committing to a specific architecture of a new system.

Evolutionary Work Breakdown Structures and Iterative Workplans

- 2nd, evolutionary WBSs are created in an incremental and iterative manner.
  - The first evolutionary WBS is typically only done for the aspects of the project understood by the analyst.
  - Later on, as the analyst understands more about the evolving development process, more details are added to the WBS.
  - This encourages a more realistic view of both cost and schedule estimation.
- 3rd, since the structure of an evolutionary WBS is not tied to any specific project, evolutionary WBSs enable the comparison of the current project to earlier projects.
  - This supports learning from past successes and failures.

Evolutionary Work Breakdown Structures and Iterative Workplans

- In the case of the enhanced Unified Process, the workflows are the major points listed in the WBS.
- Next, each workflow is decomposed along the phases of the enhanced Unified Process.
- After that, each phase is decomposed along the tasks that are to be completed to create the deliverables associated with the phase.
- The tasks listed for each workflow is dependent upon the level of activity on the workflow during each phase.
  - For example, typical activities for the inception phase of the requirements workflow would be to interview stakeholders, develop vision document, and develop use cases, while there are probably no tasks associated with the inception phase of the operations and support workflow.

Evolutionary WBS Template for the Enhanced Unified Process

- **I.** Business Modeling
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **II.** Requirements
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **III.** Analysis
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **IV.** Design
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **V.** Implementation
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **VI.** Test
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **VII.** Deployment
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **VIII.** Configuration and Change Management
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **IX.** Project Management
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **X.** Environment
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **XI.** Operations and Support
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production
- **XII.** Infrastructure Management
  - **a.** Inception
  - **b.** Elaboration
  - **c.** Construction
  - **d.** Transition
  - **e.** Production

79

Evolutionary Work Breakdown Structures and Iterative Workplans

- Using minimized approach, the first evolutionary WBS would only focus on
  - the planning step,
  - requirements-gathering and use-case development step,
  - the first build.
- After the first build is completed, a new incremental version of the WBS would be created to take into account the information that flowed from the first build back to the requirements-determination and use-case development step,
  - which, in turn, flowed back into the planning step.
- The new WBS would then address these issues and issues related to the second build.
- This iterative planning process would continue until the system was completed or the project was abandoned.

80

Evolutionary WBS Template for the MOOSAD Approach

- **1.** Planning
  - **1.1.** First Build
  - **1.2.** Second Build
  - …
  - **1.3.** Nth Build
- **2.** Requirements Gathering and Use Case Development
  - **2.1.** First Build
  - **2.2.** Second Build
  - …
  - **2.3.** Nth Build
- **3.** Builds
  - **3.1.** First Build
    - **3.1.1.** Analysis
    - **3.1.2.** Design
    - **3.1.3.** Implementation
  - **3.2.** Second Build
    - **3.2.1.** Analysis
    - **3.2.2.** Design
    - **3.2.3.** Implementation
  - …
  - **3.3.** Nth Build
    - **3.3.1.** Analysis
    - **3.3.2.** Design
    - **3.3.3.** Implementation
- **4.** Installation

81

Staffing the Project

includes determining

- how many people should be assigned to the project,
- matching people's skills with the needs of the project,
- motivating them to meet the project's objectives,
- minimizing the conflict that will occur over time.

- The deliverable for this part of project management is
  - a staffing plan,
    - which describes the number and kinds of people who will work on the project,
  - the overall reporting structure,
  - the project charter,
    - which describes the project's objectives and rules.

83

Key Definitions

- The staffing plan
  - describes the kinds of people working on the project
- The project charter
  - describes the project's objectives and rules
- A functional lead
  - manages a group of analysts
- A technical lead
  - oversees progress of programmers and technical staff members

84

Staffing Plan

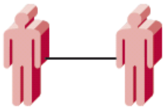
- 1^{st} step to staffing
  - determining the average number of staff needed for the project.
- To calculate this figure,
  - divide the total person-months of effort by the optimal schedule.
    - So to complete a 40 person-month project in 10 months, a team should have an average of four full-time staff members,
    - However, this may change over time as different specialists enter and leave the team (e.g., business analysts, programmers, technical writers).

85

- Adding more staff to a project to shorten the project's length does not translate into increased productivity;
  - staff size and productivity share a disproportionate relationship,
    - mainly because a large number of staff members is more difficult to coordinate.
  - The more a team grows, the more difficult it becomes to manage.

86

- For example,
  - To work on a two person project team is easy

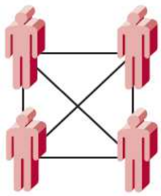


Two-person team

- the team members share a single line of communication.

87

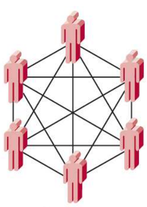
- But adding two people increases the number of communication lines to six,



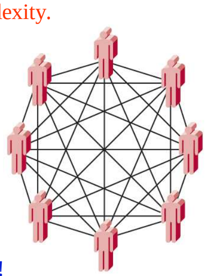
Four-person team

88

- greater increases lead to even more dramatic gains in communication complexity.



Six-person team



Eight-person team

Complexity = $n!/\{(n-r)! \times r!$

where $r = 2$

89

- One way to reduce efficiency losses on teams is
  - to understand the complexity that is created in numbers
  - to build in a reporting structure that tempers its effects.
- The rule of thumb is to keep team sizes under eight to ten people;
  - therefore, if more people are needed, create subteams.
  - In this way, the project manager can keep the communication effective within small teams,
    - which in turn communicate to a contact at a higher level in the project.

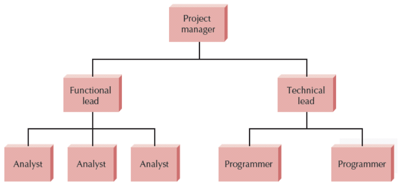
90

- After the project manager understands how many people are needed for the project,
  - (s)he creates a staffing plan
    - lists the roles that are required for the project and the proposed reporting structure for the project.
- Typically, a project will have one project manager
  - who oversees the overall progress of the development effort,
    - with the core of the team comprising the various types of analysts.
- A functional lead is assigned to manage a group of analysts,
- Technical lead oversees the progress of a group of programmers and more technical staff members.

91

Possible reporting structures

- There are many structures for project teams;
  One possible configuration of a project team of a project team:



- After the roles are defined and the structure is in place, the project manager needs to think about which people can fill each role.
  - Often, one person fills more than one role on a project team.

92

technical skills

- People have technical skills and interpersonal skills,
  - both are important on a project.
- Technical skills are useful
  - when working with technical tasks
    - e.g., programming in Java
  - in trying to understand the various roles that technology plays in the particular project
    - e.g., how a Web server should be configured on the basis of a projected number of hits from customers.

93

interpersonal skills

- Interpersonal skills include interpersonal and communication abilities that are used when dealing with
  - business users,
  - senior management executives,
  - other members of the project team.
- They are particularly critical when performing the requirements-determination activities and when addressing organizational feasibility issues.
- Each project will require unique technical and interpersonal skills.
  - For example, a Web-based project may require Internet experience or Java programming knowledge, or a highly controversial project may need analysts who are particularly adept at managing political or volatile situations.

94

- Ideally, project roles are filled with people
  - who have the right skills for the job
- However, the people who fit the roles best may not be available;
  - they may be working on other projects,
  - they may not exist in the company.
- Therefore, assigning project team members really is a combination of finding people
  - with the appropriate skill sets
  - who are available.
- When the skills of the available project team members do not match what is actually required by the project, the project manager has several options to improve the situation.

95

- People can be pulled off other projects, and resources can be shuffled around.
  - This is the most disruptive approach from the organization's perspective.
- Another approach is to use outside help—such as a consultant or contractor—to train team members and start them off on the right foot.
- Training classes are usually available for both technical and interpersonal instruction, if time is available.
- Mentoring may also be an option;
  - a project team member can be sent to work on another similar project so that (s)he can return with skills to apply to the current job.

96

Motivation

- Assigning people to tasks isn't enough;
  - Project managers need to motivate the people to make the project a success.
    - Motivation has been found to be the number-one influence on people's performance, but determining how to motivate the team can be quite difficult.
- You may think that good project managers motivate their staff by rewarding them with money and bonuses,
  - but most project managers agree that this is the last thing that should be done.
- The more often you reward team members with money, the more they expect it
  - and most times monetary motivation won't work.

97

Motivation

- Assuming that team members are paid a fair salary, technical employees on project teams are much more motivated by
  - recognition,
  - achievement,
  - the work itself,
  - responsibility,
  - advancement,
  - the chance to learn new skills.
- If you feel like you need to give some kind of reward for motivational purposes, try a pizza or free dinner, or even a kind letter or award.
  - They often have much more effective results.

98

some motivational don'ts that should be avoided to ensure that motivation on the project is as high as possible

Don'ts	Reasons
Assign unrealistic deadlines	Few people will work hard if they realize that a deadline is impossible to meet.
Ignore good efforts	People will work harder if they feel like their work is appreciated. Often, all it takes is public praise for a job well done.
Create a low-quality product	Few people can be proud of working on a project that is of low quality.
Give everyone on the project a raise	If everyone is given the same reward, then high-quality people will believe that mediocrity is rewarded—and they will resent it.
Make an important decision without the team's input	Buy-in is very important. If the project manager needs to make a decision that greatly affects the members of her team, she should involve them in the decision-making process.
Maintain poor working conditions	A project team needs a good working environment or motivation will go down the tubes. This includes lighting, desk space, technology, privacy from interruptions, and reference resources.

Source: *Adapted Rapid Development,* Redmond, WA: Microsoft Press, 1996, by from Steve McConnell.

99

Handling Conflict

- The third component of staffing is organizing the project to minimize conflict among group members.
- Group cohesiveness
  - the attraction that members feel to the group and to other members

  contributes more to productivity than do project members' individual capabilities or experiences.
- Some good measures to avoid conflict:
  - Clearly defining the roles on the project
  - Holding team members accountable for their tasks
- Some project managers develop a project charter that lists the project's norms and ground

100

Handling Conflict

- The third component of staffing is organizing the project to minimize conflict among group members.
- Group cohesiveness
  - the attraction that members feel to the group and to other members

  contributes more to productivity than do project members' individual capabilities or experiences.
- Some good measures to avoid conflict:
  - Clearly defining the roles on the project
  - Holding team members accountable for their tasks
- Some project managers develop a project charter that lists the project's norms and ground

101

Conflict Avoidance Strategies

- Some good measures to avoid conflict:
  - Clearly defining the roles on the project
  - Holding team members accountable for their tasks
  - Some project managers develop a project charter that lists the project's norms and ground rules.
    - For example, the charter may describe when the project team should be at work, when staff meetings will be held, how the group will communicate with each other, and the procedures for updating the workplan as tasks are completed
  - Develop schedule commitments ahead of time
  - Forecast other priorities and their possible impact on the project

102

Coordinating Project Activities

continues throughout the entire project until a system is delivered to the project sponsor and end users.

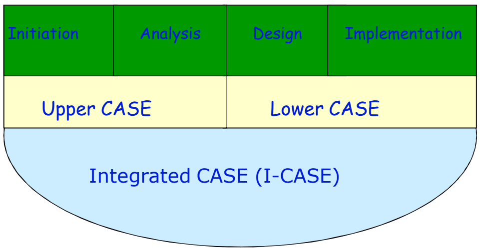
ensure that the project stays on track and that the chance of failure is kept at a minimum.

Coordinating Project Activities

- The act of coordinating project activities continues throughout the entire project until a system is delivered to
  - the project sponsor
  - the end users.
- This step includes
  - putting efficient development practices in place and mitigating risk.
- These activities occur over the course of the entire SDLC,
- but it is at this point in the project when the project manager needs to put them in place.
- These activities ensure that the project stays on track and that the chance of failure is kept at a minimum.

104

CASE Tools



105

CASE Tools

- a category of software that automates all or part of the development process.
- Some CASE software packages are primarily used during the analysis phase to create integrated diagrams of the system and to store information regarding the system components
  - often called upper CASE
- others are design-phase tools that create the diagrams and then generate code for database tables and system functionality
  - often called lower CASE

106

CASE Tools

- Integrated CASE (I-CASE) contains functionality found in both upper-CASE and lower-CASE tools in that
  - it supports tasks that happen throughout the SDLC.
- CASE comes in a wide assortment of flavors in terms of complexity and functionality
- there are many good programs available in the marketplace,
  - such as the Visible Analyst Workbench, Oracle Designer/2000, Rational Rose, and the Logic Works suite.

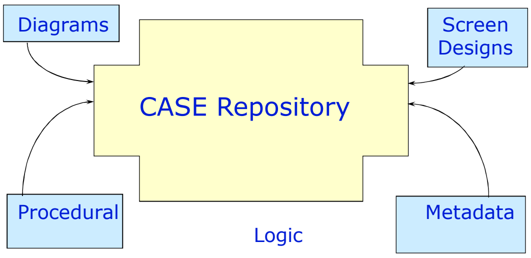
107

CASE Tools

- Benefits to using CASE:
  - tasks are much faster to complete and alter,
  - development information is centralized,
  - information is illustrated through diagrams,
    - which typically are easier to understand.
- Potentially, CASE can
  - reduce maintenance costs,
  - improve software quality,
  - enforce discipline,
- some project teams even use CASE to assess the magnitude of changes to the project.

108

CASE Components



CASE Components

- The central component of any CASE tool is
  - the CASE repository,
    - known as the information repository or data dictionary.
  - CASE repository stores the diagrams and other project information, such as screen and report designs,
  - It keeps track of how the diagrams fit together.
    - For example, most CASE tools will warn you if you place a field on a screen design that doesn't exist in your data model.
- As the project evolves, project team members perform their tasks using CASE.

Standards

- Members of a project team need to work together
- When people work together, things can get confusing.
  - For example, people sometimes get reassigned in the middle of a project.
    - It is important that their project knowledge does not leave with them and that their replacements can get up to speed quickly.
- One way to make certain that everyone is on the same page by performing tasks in the same way and following the same procedures is
  - to create standards that the project team must follow.
- Standards can range from formal rules for naming files to forms that must be completed when goals are reached to programming guidelines.

some examples of the types of standards

Types of Standards	Examples
Documentation standards	The date and project name should appear as a header on all documentation.
	All margins should be set to 1 inch.
	All deliverables should be added to the project binder and recorded in its table of contents.
Coding standards	All modules of code should include a header that lists the programmer, last date of update, and a short description of the purpose of the code.
	Indentation should be used to indicate loops, if-then-else statements, and case statements.
	On average, every program should include one line of comments for every five lines of code.
Procedural standards	Record actual task progress in the work plan every Monday morning by 10 A.M.
	Report to project update meeting on Fridays at 3:30 P.M.
	All changes to a requirements document must be approved by the project manager.

some examples of the types of standards

Specification requirement standards	Name of program to be created
	Description of the program's purpose
	Special calculations that need to be computed
	Business rules that must be incorporated into the program
	Pseudocode
	Due date
User interface design standards	Labels will appear in boldface text, left-justified, and followed by a colon.
	The tab order of the screen will move from top left to bottom right.
	Accelerator keys will be provided for all updatable fields.

Standards

- When a team forms standards and then follows them, the project can be completed faster
  - because task coordination becomes less complex.
- Standards work best when they are
  - created at the beginning of each major phase of the project
  - well communicated to the entire project team.
- As the team moves forward, new standards are added when necessary.
- Some standards are applied to the entire SDLC
  - e.g., file-naming conventions, status reporting
- others are only appropriate for certain tasks
  - e.g., programming guidelines

Documentation

- The final technique that project teams put in place
  - during the planning phase
- Includes detailed information about the tasks of the SDLC.
- Stored in project binder that contains
  - all the deliverables
  - all the internal communication that takes place
    - the history of the project.

115

Documentation

- Waiting until the last minute to create documentation
  - A poor project management practice
  - leads to an undocumented system that no one understands.
- For example, many problems that companies had updating their systems to handle the year 2000 crisis were the result of the lack of documentation.
- Good project teams learn to document the system's history as it evolves while the details are still fresh in their memory.

116

Setting up your documentation

- get some binders and include dividers
  - to separate content according to the major phases of the project.
- An additional divider for
  - internal communication, such as
  - the minutes from status meetings,
  - written standards,
  - letters to and from the business users,
  - a dictionary of relevant business terms.
- As the project moves forward,
  - place the deliverables from each task into the project binder with descriptions
    - so that someone outside of the project will be able to understand it,
  - keep a table of contents up to date with the content that is added.

117

Risk management

- the process of assessing and addressing the risks that are associated with developing a project.
- Many things can cause risks:
  - weak personnel,
  - scope creep,
  - poor design,
  - overly optimistic estimates.
- The project team must be aware of potential risks
  - so that problems can be avoided or controlled well ahead of time.

118

Risk management

- Project teams create a risk assessment, a document that
  - tracks potential risks along with an evaluation of the likelihood of the risk and its potential impact on the project .
- A paragraph or two is also included that explains potential ways that the risk can be addressed.
- There are many options:
  - risks could be
    - publicized,
    - avoided, or
    - even eliminated
  - by dealing with its root cause.

119

Sample Risk Assessment

Risk Assessment

RISK #1:	The development of this system likely will be slowed considerably because project team members have not programmed in Java prior to this project.
Likelihood of risk:	High probability of risk.
Potential impact on the project:	This risk likely will increase the time to complete programming tasks by 50 percent.

Ways to address this risk:
It is very important that time and resources are allocated to up-front training in Java for the programmers who are used for this project. Adequate training will reduce the initial learning curve for Java when programming begins. Additionally, outside Java expertise should be brought in for at least some part of the early programming tasks. This person should be used to provide experiential knowledge to the project team so that JAVA-related issues (of which novice Java programmers would be unaware) are overcome.

RISK #2: …

120

Risk management

- For example, imagine that a project team plans to use new technology but its members have identified a risk in the fact that its members do not have the right technical skills.
- They believe that tasks may take much longer to perform because of a high learning curve.
- One plan of attack could be to eliminate the root cause of the risk
  - the lack of technical experience by team members

  by finding time and resources that are needed to provide proper training to the team.

121

Risk management

- Most project managers keep abreast of potential risks, even prioritizing them according to their magnitude and importance.
- Over time, the list of risks will change as some items are removed and others surface.
- The best project managers, however, work hard to keep risks from having an impact on the schedule and costs associated with the project.

122

123