# Software Project Planning

Lecturer: Ngo Huy Bien
Software Engineering Department
Faculty of Information Technology
VNUHCM - University of Science
Ho Chi Minh City, Vietnam
nhbien@fit.hcmus.edu.vn

## Objectives

- To create a *project vision*
- To create and demonstrate a *PoC*
- To apply *wideband Delphi method*
- To create a *project schedule*

- To create a *feasibility study* report
- To create a *statement of work*
- To create a *software contract*

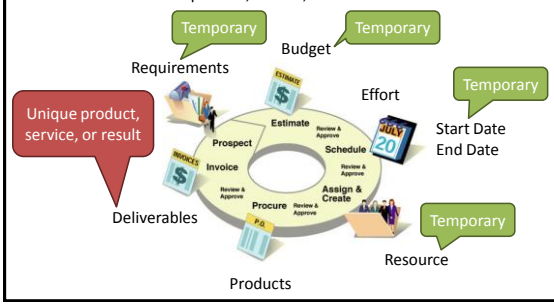
- To create a *project plan*

## References

1. Project Management Institute. A Guide to the Project Management Body of Knowledge. 4rd Edition. Project Management Institute. 2008.
2. William Trochim and James P. Donnelly. The Research Methods Knowledge Base. 2007.
3. Jennifer Greene and Andrew Stellman, Applied Software Project Management. 2005.
4. G. Goos et al.. Component-Based Software Quality - Methods and Techniques. 2003.
5. Thomas L. Saaty. How to make a decision - The Analytic Hierarchy Process. 1990.
6. Jonathan Arnowitz et al.. Effective Prototyping for Software Makers. 2007.
7. Project Management Institute. Practice Standard for Work Breakdown Structures. 2006.
8. Karl E. Wiegers. Stop Promising Miracles. 2000.
9. Project Management Institute. Practice Standard for Scheduling. 2007.
10. Jessica Keyes. Leading IT Projects - The IT Managers Guide. 2008.
11. Suzanne Robertson and James Robertson. Mastering the Requirements Process. 2nd Edition. 2006.
12. Klaus Pohl et al.. Software Product Line Engineering - Foundations, Principles, and Techniques. 2005.
13. Roger S. Pressman. Software Engineering -- A Practitioner's Approach. 7th Edition. McGraw-Hill. 2010.

## Software Project [1]

*A project is* a *temporary* endeavor undertaken to create a *unique* product, service, or result.

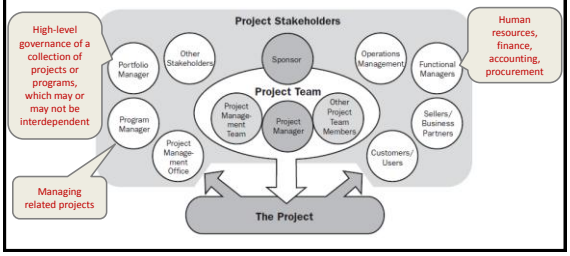



## Where Do Projects Come From? [2]

- The most common sources of projects is the experience of *practical problems* in the field.
- Another source for projects is the *literature* in your specific field.
- Another type of literature that acts as a source of good projects is the *Requests For Proposals* (RFPs) that are published by government agencies and some companies.
- Many people simply *think up* their project on their own that are influenced by their background, culture, education, and experiences.

## Project Stakeholders [1]

*Stakeholders are* persons or organizations (e.g., customers, sponsors, the performing organization, or the public), who are actively involved in the project or whose interests may be positively or negatively affected by the performance or completion of the project.



## Where to Start?



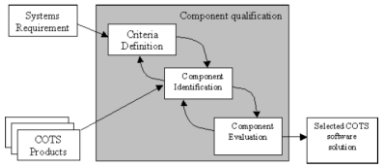

Clients

Users

Stakeholders

## COTS Evaluation [4]

- *Commercial-Off-Shelf (COTS)* software packaged solutions refer to things that one can buy, ready-made, from some manufacturer's virtual store shelf.
- *COTS software selection* is a process of determining "fitness for use" of previously-developed components that are being applied in a new system context.

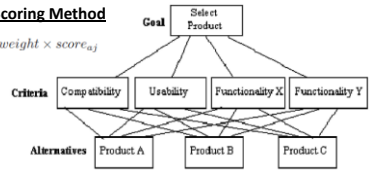



## Multi Criteria Decision Making [5]

- The basic concepts of Multi Criteria Decision Making (MCDM) approaches are establishing a set of criteria that products should meet, assigning *scores* to each criterion based on its relative importance in the decision and then ranking products based on their total scores.

**Weighted Scoring Method**

$$Score_a = \sum_{j=1}^{n} weight \times score_{aj}$$




## Analytical Hierarchy Process (AHP)

- Thomas L. Saaty. *How to make a decision - The Analytic Hierarchy Process*. 1990.
- http://en.wikipedia.org/wiki/Talk:Analytic_Hierarchy_Process/Example_Leader
- http://en.wikipedia.org/wiki/Talk:Analytic_Hierarchy_Process/Example_Car

## Can You Really Do It?

## Proof Of Concept [6]

*A proof of concept or a proof of principle is* realization of a certain method or idea(s) to demonstrate its feasibility, or a demonstration in principle.

This clearly defines *how the system must operate* within its environment.


*The purpose is to verify that some concept or theory is probably capable of being useful.*

- Storyboard Prototyping
- UI Prototyping
- Domain Prototyping
- Database Prototyping
- Architecture Prototyping
- Wireframe Prototyping

## Tools

- Balsamiq Mockups http://balsamiq.com/
- Axure RP http://www.axure.com/

## Why PoC?

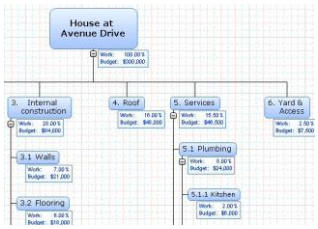
- Helps in analyzing *requirements* completely
- Helps in identifying and planning for *risks*
- Helps in making *cost/benefit* analysis
- Helps in making plan for *training* developers for implementing the system.
- Can be *reused* for actual system.


## How Much Time Do We Need?

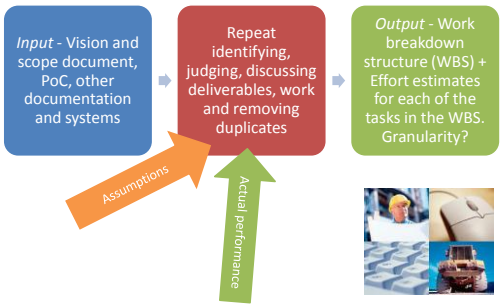
## Work Breakdown Structure [7]

*WBS is* the deliverable-oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables.

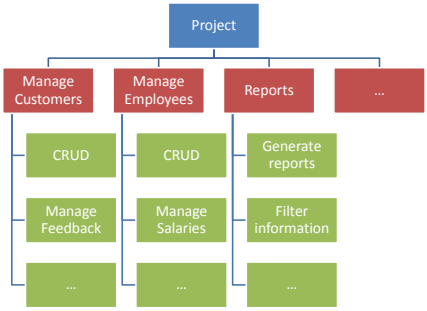
- Work. Sustained physical or mental *effort*, exertion, or exercise of skill to overcome obstacles and achieve an objective.
- Breakdown. Division into *parts* or categories; separation into simpler substances; decomposition.
- Structure. Something *arranged* in a definite pattern of organization.



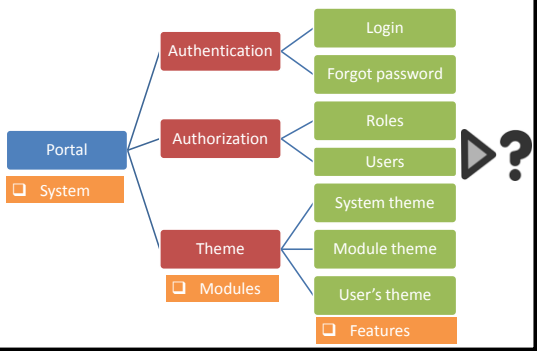

## How to Create WBS (Round 1)?




## Feature Decomposition



## Large System Decomposition

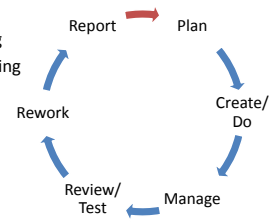



## Use Case Decomposition

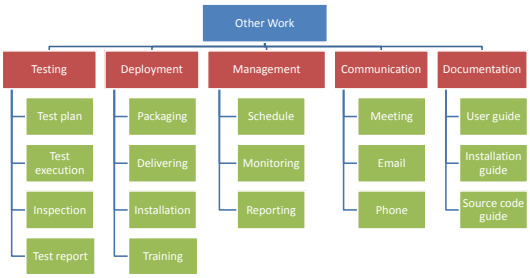
- Reading and understanding use case
- Creating database schema
- Creating UI
- Creating class diagram, sequence diagram, flow chart, state chart.
- Coding and unit testing
- System testing, bug fixes
- Documentation
- Management, communication
- Risk reserve


## Work Decomposition

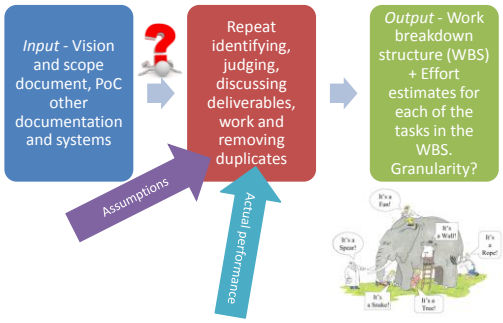
- Planning
- Requirement understanding
- Solution (steps) understanding
- Doing work
- Review and testing
- Rework
- Documentation
- Management
- Communication
- Risk reserve



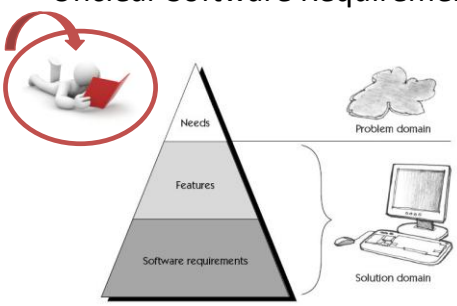

## Other Work Decomposition




## How to Create WBS (Round 2)?



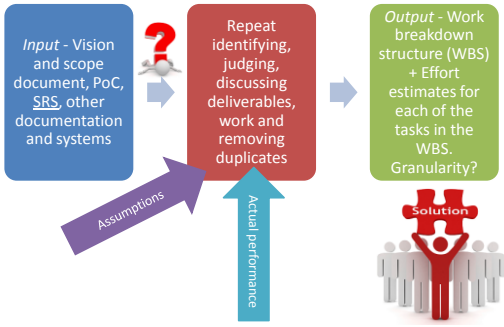

## Unclear Software Requirements



## Requirements Engineering Decomposition

- Communication
- Designing UIs
- Creating domain model
- Creating database model
- Writing software requirement specification (use cases, business rules, usability, operation, environment, security, documentation, programming languages, technologies, constrains...)
- Review and rework
- Management
- Risk reserve

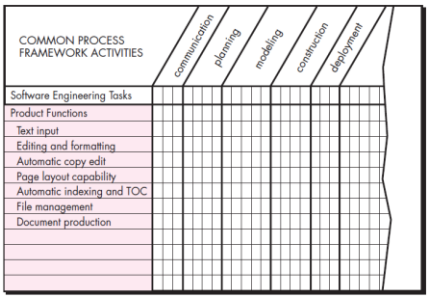

## How to Create WBS (Round 3)?




## Solution Decomposition

- Reading documents
- Understanding requirements
- Learning and investigation
- Writing design specification (architecture, system components, technologies, 3rd party components, algorithms and patterns)
- Review and rework
- Management
- Risk reserve


## Melding the Product and the Process [13]




## Task Estimation

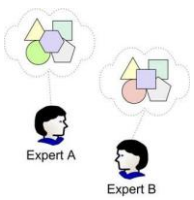
**Goal statement** To estimate the time to develop prototype for customers A & B

**Estimators** Mike, Quentin, Jill, Sophie **Units** days

*Shaded items must be discussed*

| WBS# or priority | Task name | M. | Q. | J. | S. | Best-case | Worst-case | Avg.-hi & lo | Notes |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Interview customers (A+B) | 6 | 4 | 3 | 3 | 3 | 6 | 3.5 | |
| 2 | Develop requirements docs | 5 | 10 | 2 | 5 | 2 | 10 | 5 | Discrepancy between Q. and J. |
| 3 | Inspect requirements docs | 7 | 5 | 6 | 5 | 5 | 7 | 5.5 | |
| 4 | Do rework | 8 | 7 | 9 | 7 | 7 | 9 | 7.5 | |
| 5 | Prototype design | 28 | 23 | 31 | 25 | 23 | 31 | 26.5 | |
| 6 | Test design | 9 | 7 | 6 | 6 | 6 | 9 | 6.5 | |
| | **Total** | 63 | 56 | 57 | 51 | 46 | 72 | 54.5 | |

## Who Creates the Estimates?



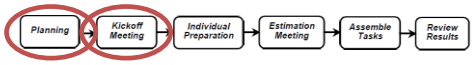

To create the task-level estimates, have the people who will actually do the work create the estimates.

Expert Estimator

Expert development, quality assurance, and documentation staff

This can be Individuals or Groups (Wideband Delphi)

## Wideband Delphi Method [8]

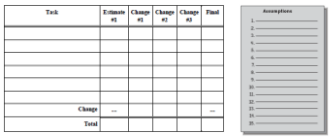



- A *problem specification*
- A *moderator*, who plans and coordinates the activity, the *project manager* and two to *four other estimators*.

- The team reviews the estimation objectives and *discusses* the problem and any estimation issues.
- The participants agree on the *estimation units*.
- All team members are *sufficiently knowledgeable* to contribute to the estimation activity.

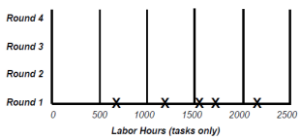
## Individual Preparation


- The estimation process begins with each participant independently developing an initial *list of the tasks* that will have to be completed to reach the stated project goal.
- Each participant then estimates the *effort* each task will consume.




## Estimation Meeting – Round 1


- The moderator begins the estimation meeting by collecting the participants' individual estimates and creating *a chart*.
- The moderator does *not identify who* created each estimate.

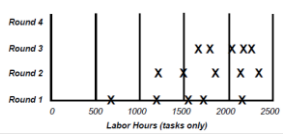



## Estimation Meeting – Discussion

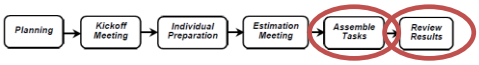

- Each estimator *reads out* his or her initial task list, identifying any assumptions made and raising any questions or issues, *without revealing* which estimate was theirs.
- Each participant will have listed different tasks that need to be performed. Combining these individual task lists leads to a more *complete list*.
- During this *initial discussion*, the team members also talk about their assumptions, estimation issues and questions they have about the problem.

## Estimation Meeting – Round 2


- All participants *modify* their estimates concurrently (and silently) in the meeting room.
- All estimators can *add new tasks* to their forms and *note any changes* they wish to make to their initial task estimates.
- The moderator *collects* the revised overall estimates and plots them on the same chart.




## Assembling Tasks and Review




- Either the moderator or the project manager assembles the project tasks and their individual estimates into a *single master task list*, merges the individual *lists of assumptions*, quality- and process-related activities, overhead tasks and wait times.
- The merging process involves removing *duplicate tasks* and reaching some *reasonable resolution* of different estimates for individual tasks.
- The estimation team *reviews* the summarized results and reaches agreement on the final outcome.

# Master Task List

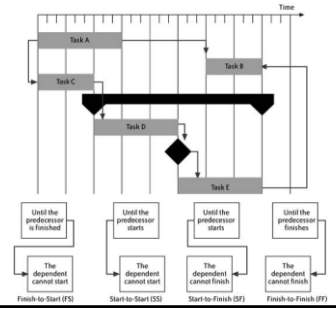
| Goal statement | To estimate the time to develop prototype for customers A & B | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Estimators | Mike, Quentin, Jill, Sophie | | | | | | | Units | days |
| | | | | | | | | *Shaded items must be discussed* | |
| WBS# or priority | Task name | M. | Q. | J. | S. | Best-case | Worst-case | Avg.-hi & lo | Notes |
| 1 | Interview customers (A+B) | 6 | 4 | 3 | 3 | 3 | 6 | 3.5 | |
| 2 | Develop requirements docs | 5 | 10 | 2 | 5 | 2 | 10 | 5 | Discrepancy between Q. and J. |
| 3 | Inspect requirements docs | 7 | 5 | 6 | 5 | 5 | 7 | 5.5 | |
| 4 | Do rework | 8 | 7 | 9 | 7 | 7 | 9 | 7.5 | |
| 5 | Prototype design | 28 | 23 | 31 | 25 | 23 | 31 | 26.5 | |
| 6 | Test design | 9 | 7 | 6 | 6 | 6 | 9 | 6.5 | |
| | Total | 63 | 56 | 57 | 51 | 46 | 72 | 54.5 | |

# Why WBS?

- *Better communication* to project sponsors, stakeholders, and team members.
- More *accurate estimation* of tasks, risks, timelines, and costs.
- Increased *confidence* that 100% of the work is identified and included.
- A foundation for the *control processes* within the project.


# Task Dependency

*Task dependency* is an indicator of the interrelationship between task start and stop times.

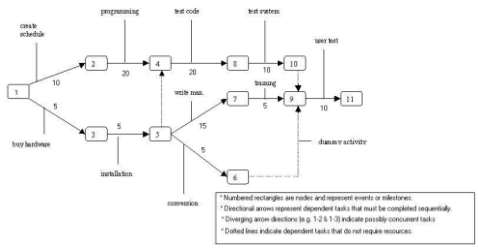



# Gantt Chart [9]

*Gantt Chart is* a chart in which activities are listed on the vertical dimension and time is shown on the horizontal dimension. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity.


# Program Evaluation and Review Technique (PERT) Chart [10]

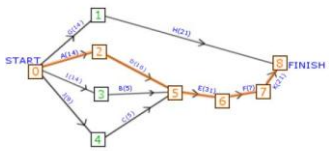
*A PERT chart presents* a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labeled vectors (directional lines) representing tasks in the project.




# Critical Path

*The critical path is made up of* a series of tasks that must finish on schedule. Any task on the critical path that will affect the project finish date if delayed.

[A] Graphics Engine
[B] Sound Engine
[C] Music Engine
[D] Input Engine
[E] Game-play/general programming
[F] Physics
[G] 2D Artwork
[H] 3D Artwork
[I] Sound effects
[J] Music recording
[K] Level Design [A] Graphics Engine



## Resource List

1. *A resource list is* a set of resources available for assignment to project tasks.
2. *A resource pool is* a resource list that can be assigned exclusively to a project or task or shared by several projects.
   - Name
   - A brief one-line description
   - Communication
   - The availability
   - The cost


## Resource Assignment

- Time (Start Date, End Date) # Effort
- Critical Path
- Time-Resource-Cost tradeoff
- Genetic Algorithms

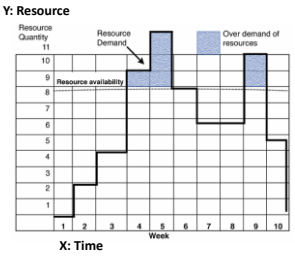

## Resource Usage

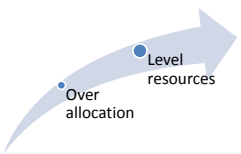
| Activity \ Week | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| A | 10 | 10 | | | | | | | |
| B | | 8 | 8 | 6 | | | | | |
| C | | | | 2 | 6 | | | | |
| D | | | | 2 | 6 | 10 | 10 | 8 | |
| E | | | | | | | | 6 | 6 |
| Total resource units requirement | 10 | 18 | 8 | 10 | 12 | 10 | 10 | 14 | 6 |

Resource unit aggregation chart: 15, 10, 5

## Resource Leveling

*Resource leveling is* a project management technique used to examine unbalanced use of resources (usually people or equipment) over time, and for resolving over-allocations or conflicts.

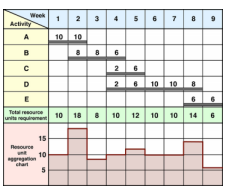



*Over-allocation means that a resource has been assigned more work than can be accomplished in the available time as dictated by the resource's calendar definition*

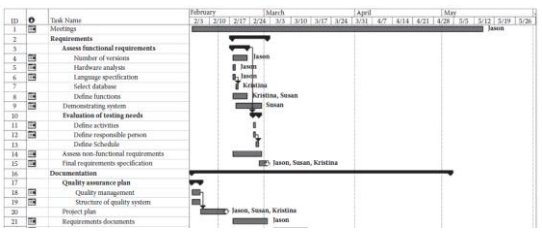
## Resource Leveling Techniques

- Break down the task into *parallel tasks* and add more resources.
- Increase *time* of a task.
- Decrease *effort* of task.
- Change parallel tasks to *sequential tasks*.
- Add *delayed time* to 2 parallel tasks.

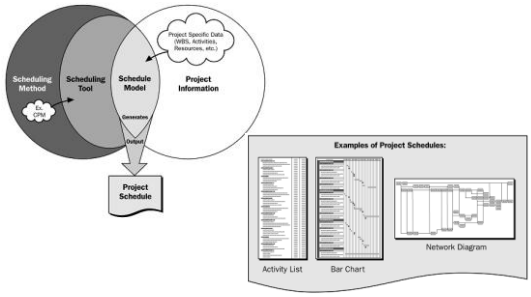


## The Project Schedule

The schedule model describes the work to be done (*what*), the resource(s) required to do it (*who*), and the *optimum sequence* (activity starts, finishes, and relationships) in which the work should be undertaken (*when*).



## How to Create Project Schedule?




## Project Estimate Summary

- Project: XYZ.
- *Start*: 04/23/14. *Finish*: 07/17/14
- *Total effort*: 720 man-days. *Duration*: 61 days. *Cost*: $21580.
- *Milestones*:
  - Milestone 1: 05/07/14: Requirements and design documents (10 days)
  - Milestone 2: 05/28/14: Test plan and module 1 (15 days)
  - Milestone 3: 06/11/14: Module 2 and module 3 (10 days)
  - Milestone 4: 07/01/14: Module 4 and module 5 (14 days)
  - Milestone 5: 07/17/14: Module 6 and User Guide (12 days)

## Budgeting

- A budget is the total sum of *all costs* of a project.
- This includes the following:
  - Hardware and software costs.
  - Travel and training costs.
  - Effort costs (the dominant factor in most projects)
    - The salaries of engineers involved in the project
    - Social and insurance costs
    - Effort costs must take overheads into account
  - Costs of building, heating, lighting
  - Costs of networking and communications
  - Costs of shared facilities (e.g. library, staff restaurant, etc.)

*There is not a simple relationship between the price charged for a system and its development costs.*

## Cost–Benefit Analysis

- *The cost-benefit analysis process compares* the costs of the system to the benefits of having that system.


- *The return on investment (ROI) is* the additional amount earned after costs are earned back.

$$\text{ROI} = \frac{(\text{Benefit} - \text{Cost})}{\text{Cost}}$$

*If an investment does not have a positive ROI, or if there are other opportunities with a higher ROI, then the investment should be not be undertaken.*

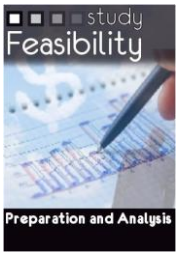
## Project Cancellation?


## Feasibility Study

*A feasibility study is* a detailed assessment of the need, value, and practicality of a proposed enterprise, such as systems development.

- *Technology* and system feasibility
- *Resource* feasibility
- *Legal* feasibility
- *Operational* feasibility
- *Schedule* feasibility
- *Market* and real estate feasibility
- *Economic* feasibility
- *Cultural* feasibility

## Feasibility Study Report




- Purpose
- Reason
- Background information
- The evaluation criteria
- The study findings
- The recommendations

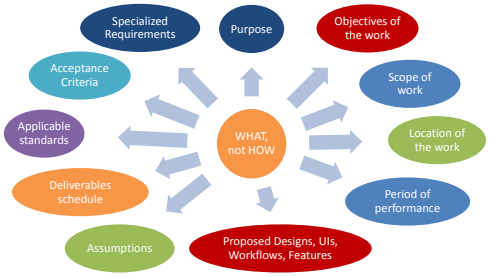
## Why Feasibility Study?


- Helps in analyzing requirements completely
- Helps in indentifying and planning for risks
- Helps in making cost/benefit analysis
- Helps in making plan for training developers for implementing the system.

## How to Make Negotiation?

## Statement of Work

*SOW is* a formal written description of your minimum requirements to be performed by a contractor.

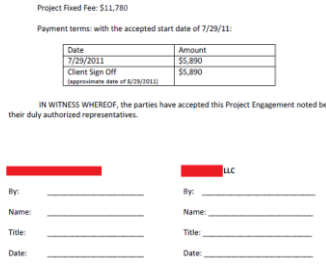



## Why SOW?


- Provides a clear understanding of the requirements
- Establishes a baseline for proposal evaluation
- Reduces evaluation and negotiation time
- Minimizes need for future changes
- Baselines contractor performance measures

## Software Contract

*Custom-software development agreement* that stipulates the rights and responsibilities of a programmer (or vendor) and a principal or customer.



Project Fixed Fee: $11,780

Payment terms: with the accepted start date of 7/29/11:

| Date | Amount |
|---|---|
| 7/29/2011 | $5,890 |
| Client Sign Off (approximate date of 8/29/2011) | $5,890 |

IN WITNESS WHEREOF, the parties have accepted this Project Engagement noted below to be executed by their duly authorized representatives.

[redacted]

By: _____
Name: _____
Title: _____
Date: _____

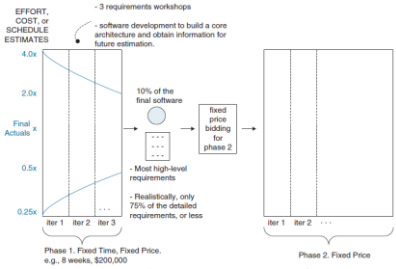
[redacted] LLC

By: _____
Name: _____
Title: _____
Date: _____

- Identification of the parties
- Payment
- Other costs
- Late fees
- Changes in project scope
- Delays
- Training
- Support and Maintenance
- Warranties
- Responsibilities

## Fixed-Price Contract

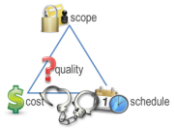

What about if requirements are too vague?




Two contract phases

## Time and Materials Contract

*An arrangement under which a contractor is paid on* the basis of (1) actual cost of direct labor, usually at specified hourly rates, (2) actual cost of materials and equipment usage, and (3) agreed upon fixed add-on to cover the contractor's overheads and profit.





- The client has to trust that the developer is spending money wisely.
- The developer has to trust that the client won't cancel the ongoing contract without good reason.

## Where Do We Go?


## Project Plan [13]

- Any complicated journey can be simplified if a map exists.
- A software project is a complicated journey, and *the planning activity creates a "map"* that helps guide the team as it makes the journey.
- *The map—called a software project plan*—defines the software engineering work by describing
  - *the work products* to be produced (aka statement of work),
  - *the technical tasks* to be conducted (aka architecture),
  - *the resources* that will be required (resource list) and,
  - *a work schedule* (project schedule), and
  - *the risks* that are likely (risk management plan).

## The W$^5$HH Principle

- *Why* is the system being developed?
- *What* will be done?
- *When* will it be done?
- *Who* is responsible for a function?
- *Where* are they located organizationally?
- *How* will the job be done *technically and managerially*?
- *How much* of each resource is needed?

## Thank You For Your Time