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

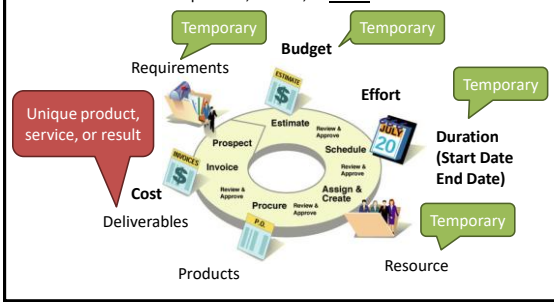
- To create an *executive summary*
- To create a *project vision*
- To create a *project charter*
- To create a *PoC*
- To create a *WBS*
- To apply *wideband Delphi* method
- To create a *project estimate*
- To create a *project schedule*
- To create a *feasibility study* report
- To create a *statement 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. 5th Edition. 2013.
2. William Trochim and James P. Donnelly. The Research Methods Knowledge Base. 2007.
3. Kathy Schwalbe. An Introduction to Project Management. Fifth Edition. 2015.
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.
14. Cincom Systems. To Build or Buy A Question of Application Development for Compliance and Quality Systems. 2008.
15. Bertrand Meyer. Object Oriented Software Construction 2nd Ed. 1997.
16. Steve McConnell. Software Estimation: Demystifying the Black Art. 2006.

What is a Project? [1]

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

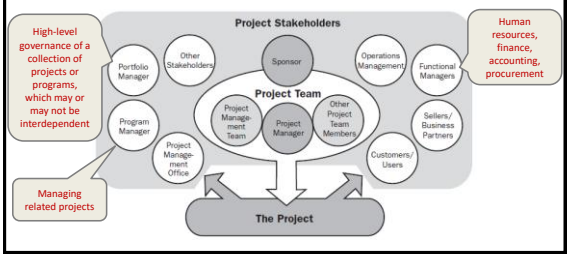


Where Do Projects Come From? [2]

- 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.

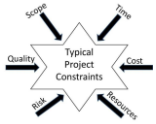
Project Stakeholders [1]

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.



Why Project Management? [3]

- Every project is constrained in *different* ways.
- You must decide which constraints are *most important* on each particular project.



Typical Project Constraints

Development is a triangle: good, fast, cheap – pick any two.

- How can you avoid the problems that occur when you meet scope, time, and cost goals, but lose sight of *customer satisfaction*?
- The answer is *good project management*, which includes more than meeting project constraints.

Where Do We Start?

1. Executive Summary
2. Project Vision
3. Project Charter
4. Requirements Specification (Product Backlog)

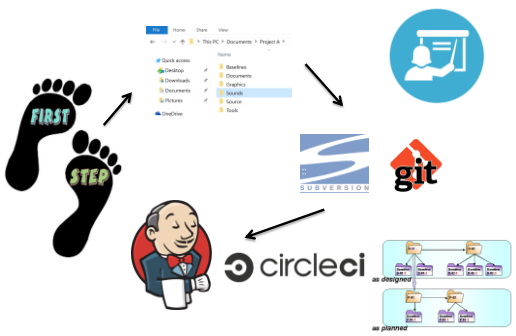
Can You Really Do It?

The Most Difficult Thing

Meeting minutes

Software Risk Management

Software Configuration Management



Building or Buying? [14]

When to Build?

- For limited, *ad hoc* applications
- If the problem is perceived as *unique* or highly *unusual*
- To solve a "stand-alone" problem that does not affect any *other area* of the business
- For a *one-time solution* with a relatively short life span
- If the organization maintains a *large* and *talented IT department* that is dedicated to remaining with the company for a long time
- *Advantages* of building
  - Complete control
  - Tailored to unique business needs
  - Ownership of the software code
  - Knowledge and expertise

When to Buy? [4, 14]

- *Commercial-Off-Shelf (COTS)* software packaged solutions refer to things that one can buy, ready-made, from some manufacturer's virtual store shelf.
- Thousands of *hours* of research and development
- Fewer "*bugs*"
- Flexibility/*adaptability*
- *Disadvantages* of buying
  - Vendor retains rights to the code
  - Product functionality determined by vendor
  - Reliance on vendor's technical support to resolve issues
  - You have to learn how to *code around* someone else's idea of best practices, and that can take time.

Again, Building or Buying?

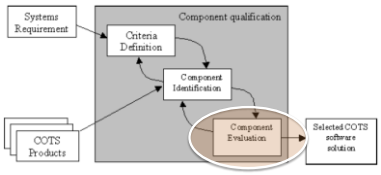
- *Re-inventing* the wheel?
- Learning from others and implementing what has been *shown to work*.

Way to Guarantee the Failure

- Buy a commercial, off-the-shelf package and *customize* it a lot.

COTS Evaluation [4]

- *COTS software selection* is a process of determining "fitness for use" of previously-developed components that are being applied in a new system context.



Buying a Mobile Phone

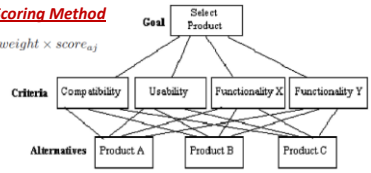
	iPhone	Samsung	LG	HTC
Display				
Battery				
Memory				
OS				
Design				
Price				

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)

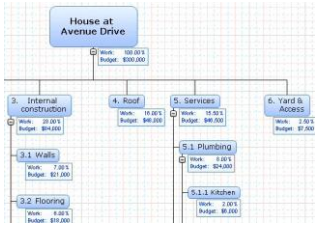
- Thomas L. Saaty. *How to make a decision - The Analytic Hierarchy Process*. 1990.
- http://en.wikipedia.org/wiki/Talk:Analytic_Hierarchy_Process/Example_Leader
- http://en.wikipedia.org/wiki/Talk:Analytic_Hierarchy_Process/Example_Car
- http://mat.gsia.cmu.edu/classes/mstc/multiple/node4.html

How Much Effort Do We Need?

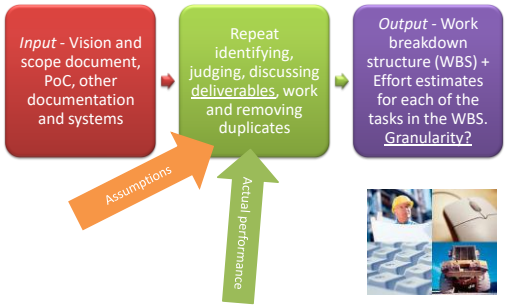
Work Breakdown Structure [7]

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.

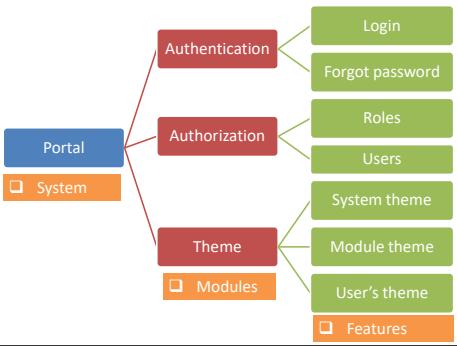
- Work. Sustained physical or mental *effort*, exertion, or exercise of skill to overcome obstacles and achieve an objective.
- Breakdown. Division into *parts* or categories; separation into simpler substances; decomposition.
- Structure. Something *arranged* in a definite pattern of organization.



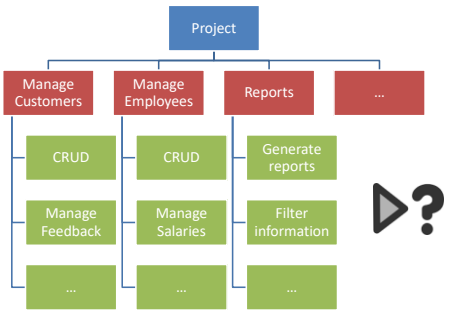
How to Create WBS (Round 1)?



System Decomposition



Features Decomposition



Use Case Decomposition

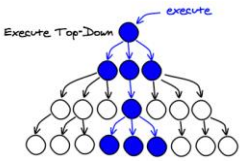
- Reading and understanding use case
- Creating *database* schema
- Creating *UI*
- Creating class diagram, *sequence diagram*, *flow chart*, state chart, algorithm description, solution description
- Coding and unit testing
- System *testing*, bug fixes
- Documentation
- Management, communication
-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

Work Decomposition

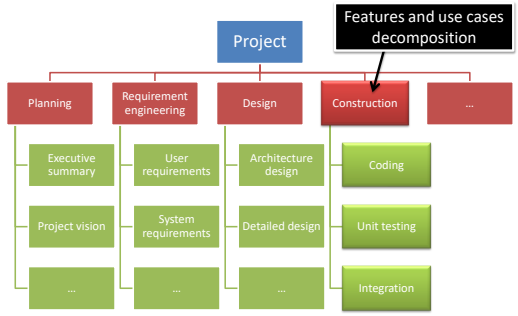
- Planning
- Requirement understanding
- Solution (steps) understanding
- Doing work
- Review and testing
- Rework
- Documentation
- Management
- Communication
- Risk reserve

Top-Down vs. Bottom-Up [15]

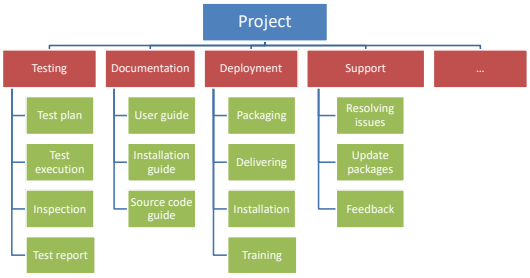
- Top-down is a reasonable way of *describing* things which are already fully understood.
- When the top-down phase begins, the *problem* is already solved, and only details remain to be solved.



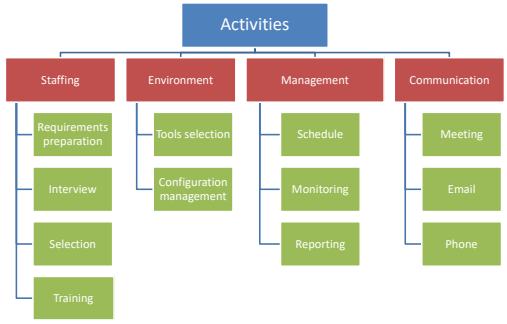
Phases Decomposition



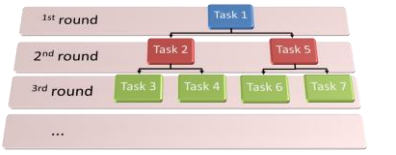
Phases Decomposition (cont.)

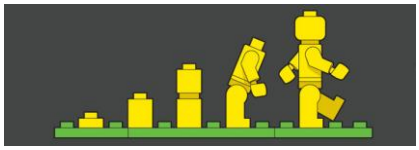


Common Activities Decomposition

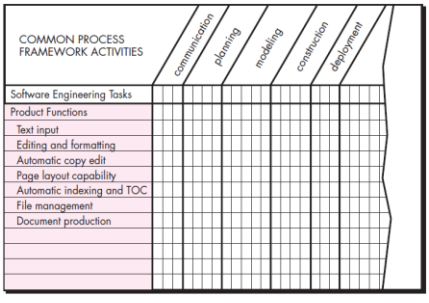


Again, Top-Down vs. Bottom-Up





Melding the Product and the Process [13]



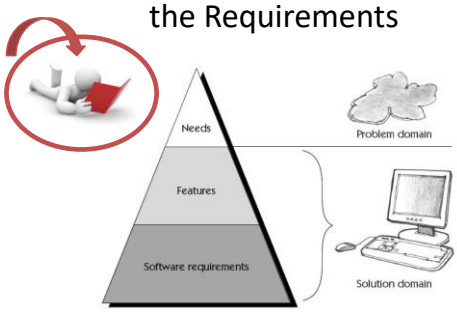
How to Create WBS (Round 2)?

Input - Vision and scope document, PoC other documentation and systems

Output - Work breakdown structure (WBS) + Effort estimates for each of the tasks in the WBS. Granularity?

Unclear Software Requirements!

Estimating Time for Understanding the 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)?

Input - Vision and scope document, PoC, SRS, other documentation and systems

Output - Work breakdown structure (WBS) + Effort estimates for each of the tasks in the WBS. Granularity?

Unclear Solution!

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

Task Estimation

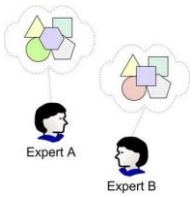
Goal statement To estimate the time to develop prototype for customers A & B

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?

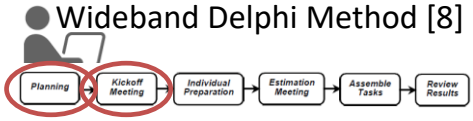


To create the task-level estimates, have the people who will actually do the work create the estimates.

Expert Estimator

Expert development, quality assurance, and documentation staff
This can be Individuals or Groups (Wideband Delphi)

Wideband Delphi Method [8]

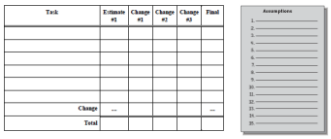


- A *problem specification*
- A *moderator*, who plans and coordinates the activity, the *project manager* and two to *four other estimators*.

- 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.

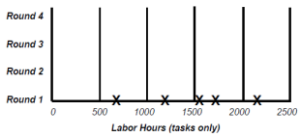
Individual Preparation

- 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.

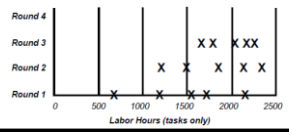


Estimation Meeting – Discussion

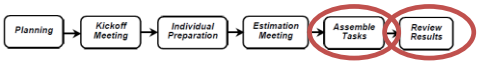
- Each estimator *reads out* his or her initial task list, identifying any assumptions made and raising any questions or issues, *without revealing* which estimate was theirs.
- Each participant will have listed different tasks that need to be performed. Combining these individual task lists leads to 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

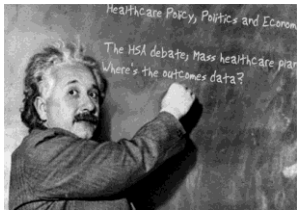
WBS# or priority	Task name	M.	Q.	J.	S.	Best-case	Worst-case	Avg.-hi & lo	Notes
1	Interview customers (A+B)	6	4	3	3	3	6	3.5	
2	Develop requirements docs	5	10	2	5	2	10	5	Discrepancy between Q. and J.
3	Inspect requirements docs	7	5	6	5	5	7	5.5	
4	Do rework	8	7	9	7	7	9	7.5	
5	Prototype design	28	23	31	25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

How to Create an Estimate Within a Very Limited Time Frame?

Expert Opinion [16]



Example of Individual Estimation

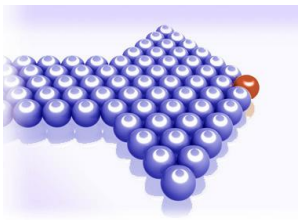
Feature	Estimated Days to Complete			
	Best Case	Most Likely Case	Worst Case	Expected Case
Feature 1	1.25	1.5	2.0	1.54
Feature 2	1.5	1.75	2.5	1.83
Feature 3	2.0	2.25	3.0	2.33
Feature 4	0.75	1	2.0	1.13
Feature 5	0.5	0.75	1.25	0.79
Feature 6	0.25	0.5	0.5	0.46
Feature 7	1.5	2	2.5	2.00
Feature 8	1.0	1.25	1.5	1.25
Feature 9	0.5	0.75	1.0	0.75
Feature 10	1.25	1.5	2.0	1.54
TOTAL	10.5	13.25	18.25	13.62

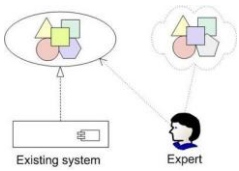
Pros & Cons of Expert Opinion

Can be used in a new business area, new technology, or a brand-new kind of software

- Cannot be quantified
- Requires experts

Analogy





Step 1. Get Detailed Size, Effort, and Cost Results for a *Similar* Previous Project

Old Project		
Database	5,000 lines of code (LOC)	10 tables
User interface	14,000 LOC	14 Web pages
Graphs and reports	9,000 LOC	10 graphs + 8 reports
Foundation classes	4,500 LOC	15 classes
Business rules	11,000 LOC	???
TOTAL	**43,500 LOC**	

Step 2. Compare the *Size* of the New Project to a Similar Past Project

Subsystem	Actual Size of Old Project	Estimated Size of New Project	Multiplication Factor
Database	10 tables	14 tables	1.4
User interface	14 Web pages	19 Web pages	1.4
Graphs and reports	10 graphs + 8 reports	14 graphs + 16 reports	1.7
Foundation classes	15 classes	15 classes	1.0
Business rules	???	???	1.5

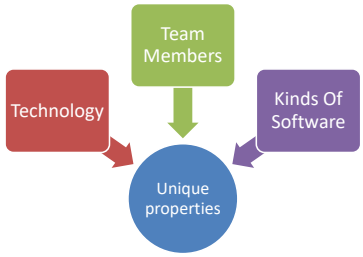
Step 3. Build Up the *Estimate* for the New Project's Size as a Percentage of the Old Project's Size

Subsystem	Old Project	Multiplication Factor	New Project
Database	5,000	1.4	7,000
User interface	14,000	1.4	19,600
Graphs and reports	9,000	1.7	15,300
Foundation classes	4,500	1.0	4,500
Business rules	11,000	1.5	16,500
TOTAL	**43,500**	-	**62,900**

Step 4. Create an *Effort Estimate* Based on the Size of the New Project Compared to the Previous Project

Term	Value
Size of New Project	62,900 LOC
Size of Old Project	÷ 43,500 LOC
Size ratio	**= 1.45**
Effort for Old Project	× 30 staff months
Estimated effort for New Project	**= 44 staff months**

Step 5. Check for Consistent *Assumptions* Across the Old and the New Project



Pros & Cons of Analogy

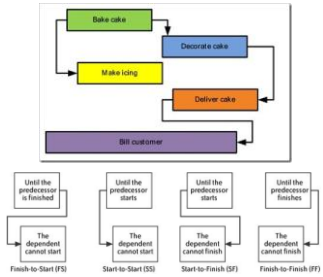
- ✓ Simple, accurate, cheap
- ✓ Based on proven characteristics

- ○ Impossible if no comparable project has been tackled.
- ○ The need to determine the most important variables to be used for describing the solution.

How Much Time Do We Need?

Task Dependency

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

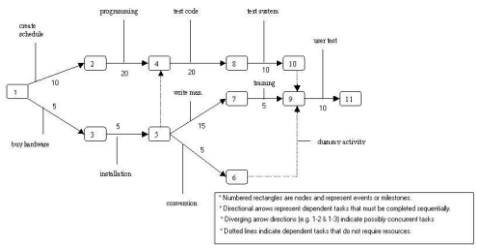


Gantt Chart [9]

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]

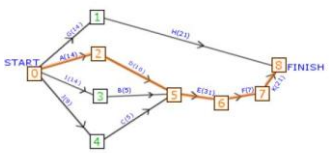
A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labeled vectors (directional lines) representing tasks in the project.



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

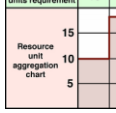
1. A *resource list* is a set of resources available for assignment to project tasks.
2. A *resource pool* is a resource list that can be assigned exclusively to a project or task or shared by several projects.
   - Name
   - A brief one-line description
   - Communication
   - The availability
   -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

Resource Assignment

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

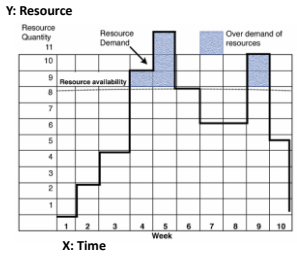
Resource Usage

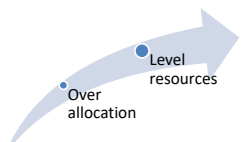
Activity \ Week	1	2	3	4	5	6	7	8	9
A	10	10							
B		8	8	6					
C				2	6				
D				2	6	10	10	8	
E								6	6
Total resource units requirement	10	18	8	10	12	10	10	14	6



Resource 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.

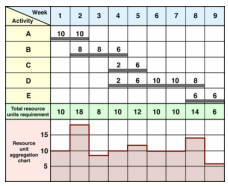




Over-allocation means that a resource has been assigned more work than can be accomplished in the available time as dictated by the resource's calendar definition.

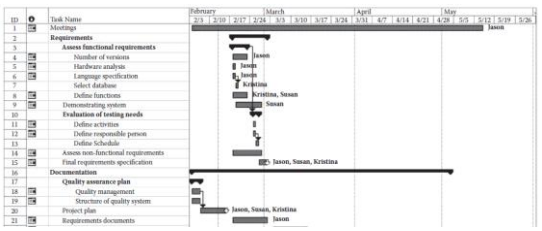
Resource Leveling Techniques

- 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 [9]

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*).

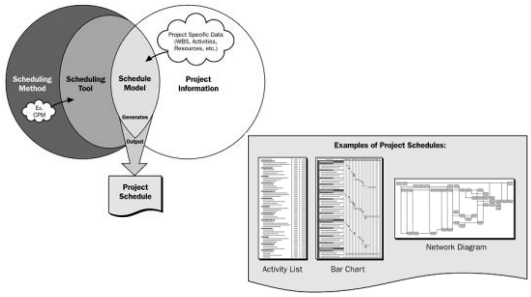


Tools

Microsoft Project 2016

- https://products.office.com/en-au/Project/project-for-office-365
- http://www.ganttproject.biz/
- https://ganttpro.com/
- https://www.smartsheet.com/

How to Create Project Schedule?



Costs and Budget

- A *budget* is the total sum of *all costs* of a project.
- This includes *the following*:
  - **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
  - Hardware and software costs.
  - Costs of building, heating, lighting, networking and communications
  - Travel and training costs
  - 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.

Benefits, Revenue and Income

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 Estimate Summary

- Project: XYZ.
- *Start*: 04/23/14. *Finish*: 07/17/14
- *Total effort*: 720 man-days. *Duration*: 61 days. *Cost*: $21580.
- Please see *attached schedule* for details.
- *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)

Project Feasibility?

Feasibility Study

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

- *Legal* feasibility ATTENTION!
- *Market* feasibility (supply and demand)
- *Economic* feasibility
- *Technology* and system feasibility
- *Resource* feasibility
- *Operational* feasibility
- *Schedule* 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

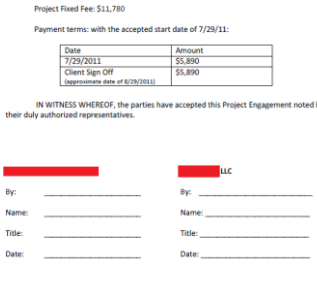
SOW is a formal written description of your minimum requirements to be performed by a contractor.

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] LLC

By: _____ By: _____

Name: _____ Name: _____

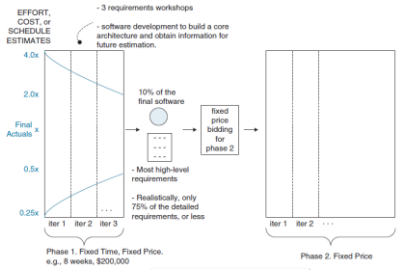
Title: _____ Title: _____

Date: _____ 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 From Here?

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 PoC, 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? (reasons and benefits)
- *What* will be done? (objectives)
- *When* will it be done? (milestones and timeline)
- *Who* is responsible for a function? (responsibility)
- *Where* are they located organizationally?
- *How* will the job be done *technically* and *managerially*? (management and technical strategy)
- *How much* of each resource is needed? (estimation)

"It is applicable regardless of size or complexity of software project."

Thank You for Your Time