

# L5DC Computing Project

## Project proposal

Achyut Timsina

Softwarica college of IT & E-Commerce  
Kathmandu, Nepal  
achyut@gmail.com

April 5, 2016

# Tasks overview

The project is divided into a number of sub-task as follows:

1. Project proposal
2. Analysis specification
3. Design specification
4. Midway presentations
5. Final report
6. Final presentation/demonstration

# Assignment rules

- ▶ All reports (doc/pdf) should be named as batch\_firstName\_lastName\_taskName\_version, For example: 9\_ram\_shrestha\_proposal.pdf(doc)
- ▶ Plagiarism will not be tolerated, and you have to submit your report to [www.turnitin.com](http://www.turnitin.com) for the plagiarism check!
- ▶ Use Harvard style for referencing.
- ▶ All assignments hard copy and soft copy should be submitted with in respective deadlines.
- ▶ The deadlines are strict!
- ▶ The deadlines are published at computing project's site.
- ▶ Failure to submit assignments on the published deadlines will result in failing the course!
- ▶ Goal is to make you working continuously towards completion of projects on time with vibrant results!
- ▶ Email address for all of your digital communications is **cpsoftwarica@gmail.com**. I will try to write back with in 24 hours.

# Project selection

- ▶ Most difficult stages of all
- ▶ Ideally have a real client for the software project
- ▶ (If you do not have real client, ) Try to be as practical as possible! (Try to solve practical problems)
- ▶ Your motivation level differs whether you have real client/project or not!
- ▶ You can look up the potential list of topics for projects.
- ▶ No two students can have exactly the same computing project topic.

# Project Scoping

Project scoping report should consist of:

**Background** : What your project is about? Describe the area/domain of the project. Provide background and description of related concepts in your project. This section should set the scene for the project. What problem you are trying to solve? Why solving those problems matter? How you are going to solve identified problem(s)?

**Main features** : What are the main functionalities that your project going to have? What will be your choice of programming language(s), tools, libraries for developing this project?

# Scoping (contd.)

**Aims and Objectives** : State a few major goals/aims of your project. For example: *To build desktop application for managing student information system of ABC college* *To make the student information more accessible at ABC college*

Your objectives should be SMART:

- ▶ Specific
- ▶ Measurable
- ▶ Appropriate
- ▶ Realistic
- ▶ Time-related

## Scoping (contd.)

State objectives that corresponds to the stated aim(s). Some of your objective can be academic, technical as well. For example:

- ▶ To analyze the student information management system
- ▶ To design the ...
- ▶ To develop ...
- ▶ To test ...
- ▶ To document ...
- ▶ To report ...
- ▶ To apply test driven development (TDD) ...

Note: Refer to 'sample\_proposal.pdf' at the FTP server!

# Project Planning

Project planning includes:

**Work breakdown structure (WBS)** produce a model that shows the tasks that must be completed in order to complete the project

**Time estimates** produce estimates of the time required to complete the tasks

**Milestones** Significant point in your project that can be used to give an indication of how well the project is going

**Activities** Identify the order in which tasks must be completed

**Present your plan** Use the tasks, time estimates, milestones and activities to produce a plan, normally in the form of a Gantt chart

**Re-plan** if required adapt your plan to changing situation



# Work breakdown structure (WBS)

- ▶ WBSs are used to break project down into lower and lower levels of details to reveal exactly what work will be done to complete the project.
- ▶ Begin WBS by breaking your projects down into its main objectives (from aims/objectives)
- ▶ Continue breaking down the objectives down into lower and lower level of detail
- ▶ General guideline is to break the activities such a that the effort required to finish the activity is not less than 5% of the total project effort.
- ▶ For a six month project, a WBS activity should not be less than one week of the time
- ▶ The total project management activity should not be greater than 10% of the total project's effort.
- ▶ for a six month project, the total project management effort should not be greater than 2 weeks

# Time estimates

- ▶ Produce estimates of the time required to complete the tasks
- ▶ Fix deadline (April 15, 2015!)
- ▶ By breaking down into smaller activities, the time estimation will be more accurate (explain with example)
- ▶ Be flexible to adapt your time estimates based on the fixed delivery date.
- ▶ add up if conservative, reduce if lax (but your deadline is fixed)!!

# WBS and Time estimates

**Table:** Work breakdown structure with time estimate

<b>WBS #</b>	<b>Task name</b>	<b>Days</b>
1.	Your project name	66
1.1	Project Management	15
1.1.1	Scoping	5
1.1.2	Planning	5
1.1.3	Monitoring & Controlling	5
1.2	Analysis	15
1.2.1	Requirements	5
1.2.2	Use cases	5
1.2.3	Architecture	5
1.3	Design	15
1.4	Testing	
1.5	Reporting	

# Identify milestones

- ▶ Milestones are significant steps towards the completion of a project.
- ▶ Help us to measure progress by providing intermediate reference points.
- ▶ Milestones can be identified by looking into the WBS!
- ▶ Milestones have a fix delivery date!
- ▶ Milestones are not activities, they are result or output of a activity.
- ▶ Refer to the milestones sections in the sample\_proposal report

# Activity sequencing

- ▶ Activity sequencing identifies the order in which the tasks should be performed.
- ▶ Also referred to as PERT networks, or network diagrams.
- ▶ Activity-on-nodes vs Activity on-arrow networks
- ▶ But we will use activity-on-nodes network diagram

# Scheduling

- ▶ Gantt charts!
- ▶ Show the durations of activities and identify instances when tasks are performed simultaneously
- ▶ Project activities are represented by rectangles or nodes
- ▶ The length of the rectangle shows the duration of the activity
- ▶ Milestones are represented by diamonds

# Gantt Chart

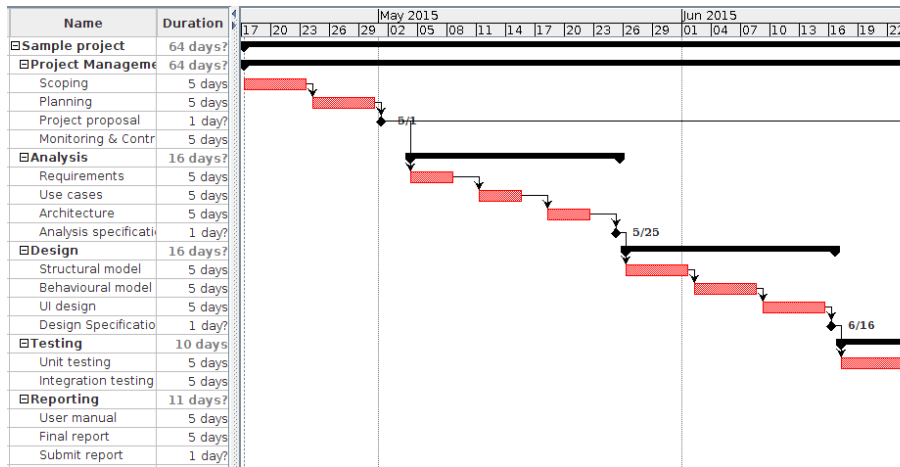


Figure: Sample Gantt chart

# Re-planning (if required!)

- ▶ Going back to your plan and making appropriate adjustments
- ▶ Should not be more than 10% of your project's total effort



# Development methods

You have to pick one suitable methodology for your project and motivate it.

1. Waterfall
2. Agile

# Risk management

- ▶ Risk management is a important aspect of project management
- ▶ The goal risk management is to ensure that you are in a position to deal with these risk if they do occur and you are not facing them ill-prepared
- ▶ Major four stages of managing and controlling risks in your project are:
  - Identify risks
  - Assess impact of risks
  - Alleviate critical risks
  - Control risks

# 1. Identify risks

Risks can be of different types:

- ▶ Event-driven risks (hard disk crashing, etc.),
- ▶ Long term or chronic risks (bad estimation, etc.),
- ▶ Technical risks (unavailability of a particular library functions, etc.)
- ▶ Non-technical risks (illness, etc.)

## 2. Assess impact of risks

- ▶ To estimate risks impact to your project, you can use following equation:  $Impact = Likelihood * Consequence$
- ▶ Most serious risks are those which are highly likely to occur and have significant consequences

# Risk likelihood and consequence scores Dawson [2005]

Table: Risk likelihood

<b>Risk likelihood</b>	<b>Score</b>
Low	1
Medium	2
High	3

Table: Risk consequence

<b>Risk consequence</b>	<b>Score</b>
Very low	1
Low	2
Medium	3
High	4
Very high	5

### 3. Alleviate critical risks

- ▶ Three ways to deal with identified risks:

**Avoidance** Reducing the chance of risk occurrence at all.

**Deflection (transfer)** Passing the risk onto something or someone else.

**Contingency** Accept that risk will occur and put something in place to deal with it when it does occur.

## 4. Control risks

- ▶ You can not just close your eyes now!
- ▶ Continuously monitor the risks as your project progresses and be prepared to deal with them.

# Configuration management

- ▶ Common problems in software development projects are in the co-ordination and control associated with project artefacts.
- ▶ CM is used to preserve the old versions of the system
- ▶ To store artefacts efficiently
- ▶ A simple directory structure for configuration management is suggested!
- ▶ See the student's guide exercise 1!



# References

Dawson [2005], Weaver [2004]

C.W. Dawson. *Projects in Computing and Information Systems: A Student's Guide*. Addison-Wesley, 2005. ISBN 9780321263551.

P.L. Weaver. *Success in your project: a guide to student system development projects*. Prentice Hall, 2004. ISBN 9780273678090. URL <http://books.google.com.np/books?id=LKZQAAAAMAAJ>.