
CIS 422/522

Review of Life Cycle Models
Project 1 - Mini SRS/SDS/Project Plan
Project Planning Tools

Life-Cycle Review (1)

- Life-cycle models provide a way to organize development activities, used to
 - Standardize development processes and products
 - Guide development
 - Track progress
 - Coordinate expectations
- Academic expectations
 - Be familiar with common models
 - Understand what development issue each model seeks to address (next lecture)

Life-Cycle Review (2)

- Real development
 - Understand that models are necessarily idealized abstractions of real processes
 - That it is useful to have a model (for reasons stated)
 - Chose which model you use based on project characteristics and organizational constraints
- Project specific
 - Identify project goals, constraints, risks
 - Choose and tailor model to address key risk issues

Project 1 Dates and Deliverables

Project 1 Dates

- Monday, April 11th: Draft project documents due for discussion on Wednesday.
 - You can give me documents earlier if you have them ready.
- Wednesday, April 13th: Teams meet with me to discuss progress, mid-course corrections
 - No class
 - Schedule a meeting time for each team
 - Present project goals and plan for discussion
- Friday, April 22nd:
 - Project deliverables due
 - In class presentation of results, lessons learned.

Mini SRS/SDS/Project Plan

- Answers Three Questions
- What does the system do?
 - Concept document (mini Concept of Operations document)
 - Requirements analysis (a mini SRS).
- How will the system work?
 - Design overview (a mini SDS) & Rationale
- How will we build it?
 - Project plan

What is a Concept Document?

- Several different types
 - Concept of Operations (ConOps)
 - Mission Requirements Document
 - Business Requirements Document
- What these have in common
 - **Audience:** stakeholders who are not software experts (but may be domain experts)
 - **Purpose:** Justifies why we are building this particular system by linking *product capabilities* to *organizational goals*
 - **Approach:** Describes the system capabilities from the *user's* point of view
 - E.g. use cases
 - Necessarily an informal specification

What is an SRS?

- The Software Requirements Specification
 - Audience: stakeholders with technical expertise in software
 - Includes: designers, testers, maintainers, coders, project management
 - May include customers, regulators and other stakeholders depending on context
 - Purpose: characterize precisely what is to be built
 - Detailed black box behavior of the system used as design-to, test-to specification.
 - Any constraints on design or implementation
 - Approach: precise, detailed and unambiguous specification to the extent possible (usually not achieved).

Mini SRS for Project

- Combines purposes of ConOps and SRS
- Problem Statement: What problem does it solve?
 - Independent of the solution.
 - Who are the users? How will they use it?
- Scenarios
 - Including mockups of external design
- Specification
 - Precise description of software behavior as possible
 - Generalize from the scenarios
 - Try to answer the hard questions

Mini SDS: Design

- Architectural design overview
 - What are the major parts?
 - What are the interfaces?
 - What are their dependencies (relationships)?
- Key design decisions
 - Which decisions drove the decomposition
 - What is the rationale for key decomposition decisions?
 - E.g., How did the decomposition address risks like the need for increments

Mini Project Plan

- Include each of these four sections:
 - Project organization: People and roles
 - Risk analysis, risk reduction strategies.
 - Work breakdown (>10 milestones) and project schedule (who will do what)
 - Monitor and report progress:
 - Keep a record of: Each task, assigned to whom, when assigned, due date, when completed, who did it, who signed off on it.

How many pages?

- My best guess:
 - 5 pages each for requirements and design, plus 3-5 pages of diagrams
 - 3 pages for plan
- What matters is content, not length

Example

SOFTWARE REQUIREMENT SPECIFICATION	1
PURPOSE OF THE DOCUMENT.....	1
PROBLEM STATEMENT	1
PROPOSED SOLUTION.....	1
USER CHARACTERISTICS.....	2
USER SCENARIOS	3
FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	4
OTHER REQUIREMENTS:	5

SOFTWARE DESIGN SPECIFICATIONS	6
DESIGN: HIGH-LEVEL VIEW.....	6
DESIGN PHASES	6
SYSTEM ARCHITECTURE	7
STRUCTURE: LOW-LEVEL VIEW OF INDIVIDUAL COMPONENTS.....	8
DESIGN REQUIREMENTS	10
ORGANIZATION PRINCIPLES.....	11
RATIONALE.....	12

PROJECT PLAN.....	12
INTRODUCTION.....	12
PROJECT ORGANIZATION.....	12
RISK ANALYSIS AND RISK REDUCTION STRATEGIES.....	13
WORK BREAKDOWN AND PROJECT SCHEDULE.....	14
SOFTWARE PROCESS MODEL.....	15
MONITOR AND REPORT PROGRESS.....	16

Project Plan

View of SE in this Course

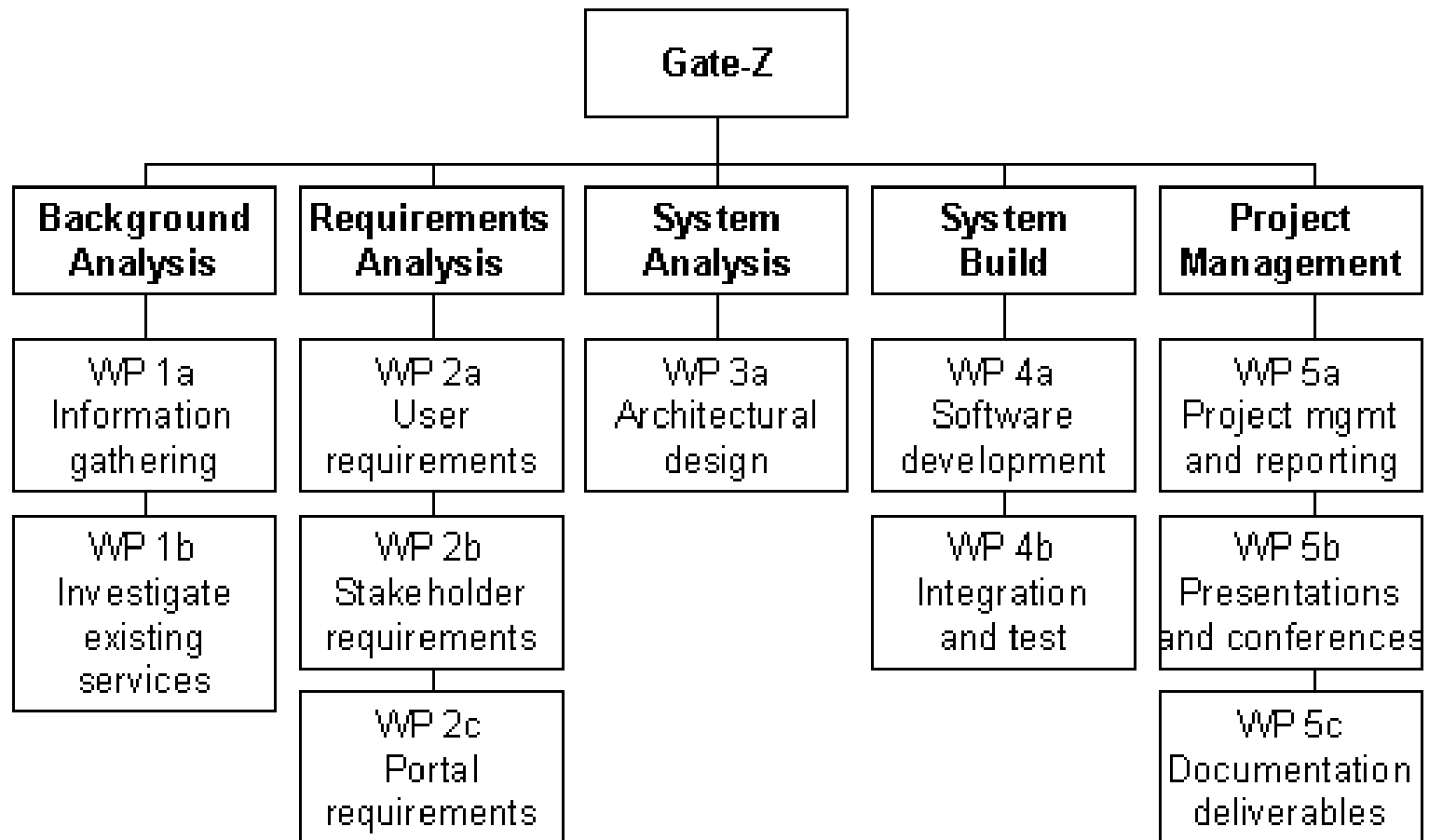
- The purpose of software engineering is to *gain* and *maintain* intellectual and managerial control over the products and processes of software development.
- “Intellectual control” means that we are able make rational choices based on an understanding of the downstream effects of those choices (e.g., on system properties).
- **Managerial control means we control development *resources* (budget, schedule, personnel).**

What are your resources?

Work Breakdown Structure

- This is a technique to analyze the content of work and cost by breaking it down into its component parts. It is produced by :-
 - Identifying the key elements
 - Breaking each element down into component parts
 - Continuing to breakdown until manageable work packages have been identified. These can then be allocated to the appropriate person
- The WBS is use to allocate responsibilities

Work Breakdown Structure



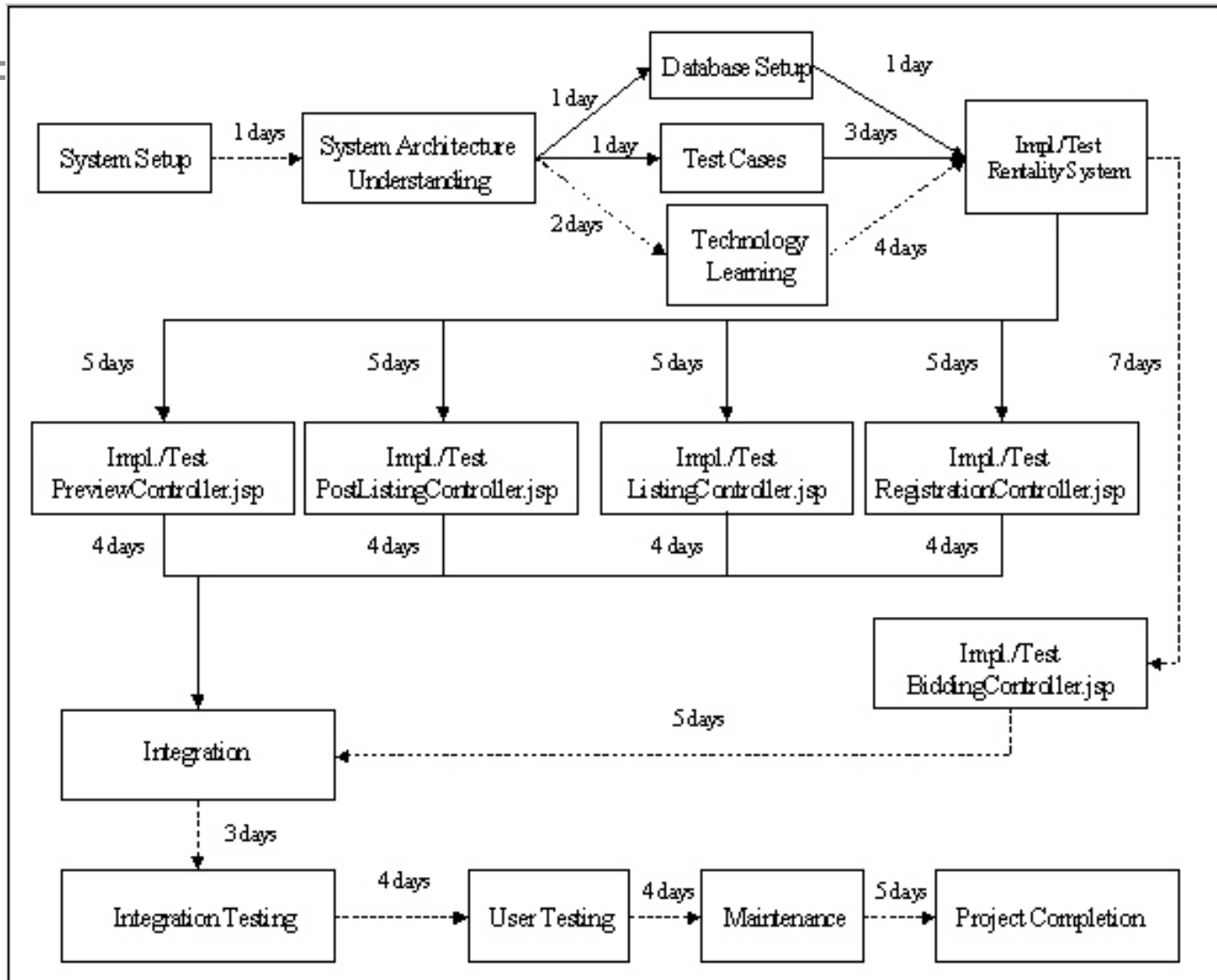
Milestone Planning

- Milestone planning is used to show the major steps that are needed to reach the goal on time
- Milestones typically mark completion of key deliverables or establishment of baselines
 - Baseline: when a work product is put under configuration management and all changes are controlled
- Often associated with management review points
- E.g., Requirements baseline, project plan complete, code ready to test

Pert Chart

- Network analysis or PERT is used to analyze the inter-relationships between the tasks identified by the work breakdown structure and to define the dependencies of each task
- Helps identify where ordering of tasks may cause problems because of precedence or resource constraints
 - Where one person cannot do two tasks at the same time
 - Where adding a person can allow tasks to be done in parallel, shortening the project

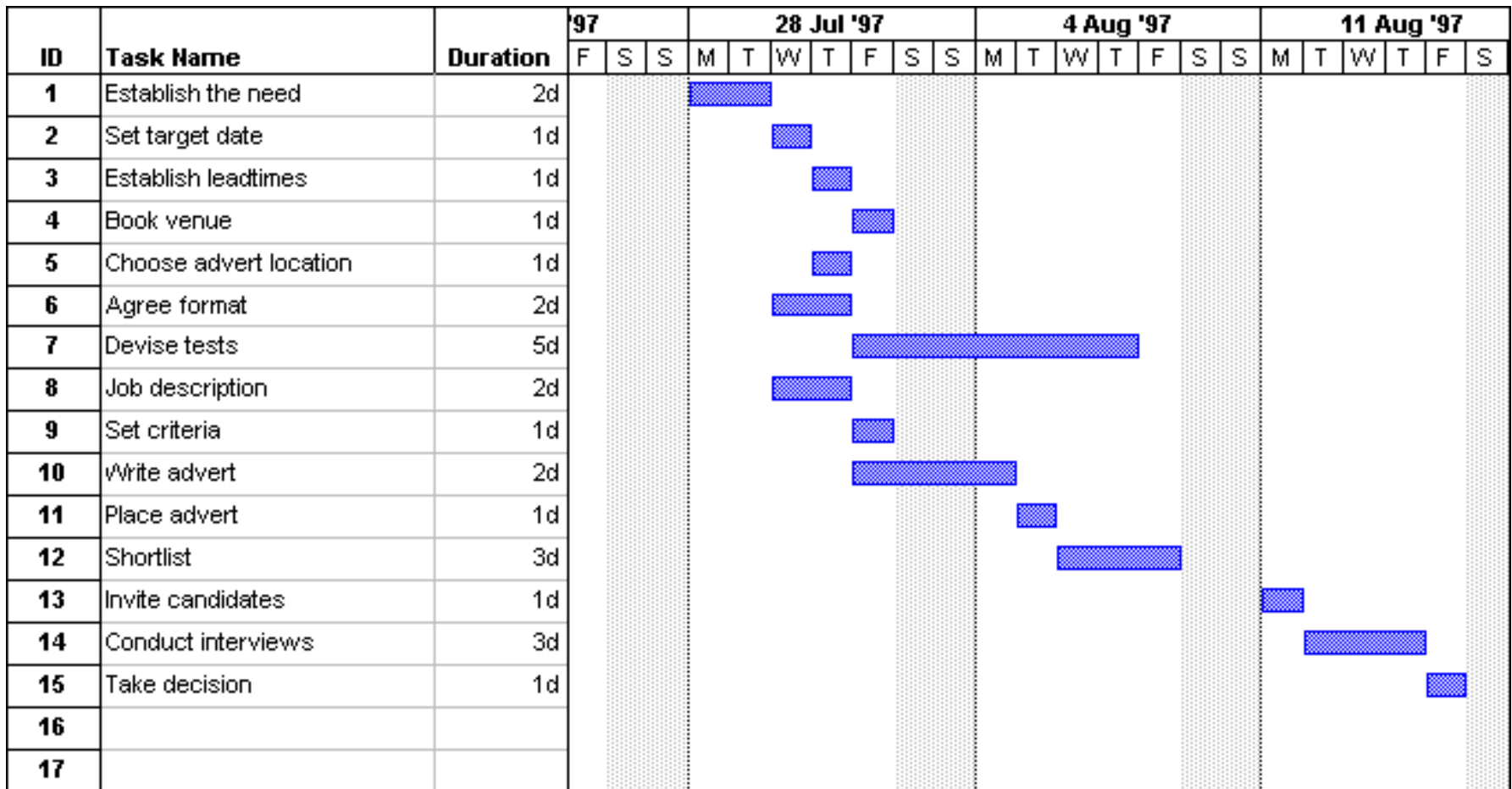
Example



Gantt Charts

- Method for visualizing a project schedule showing
 - The set of tasks
 - Start and completion times
 - Task dependencies
 - Responsibilities
- PERT charts can be reformatted as Gantt charts

Example Gantt Chart



<http://www.spottydog.u-net.com/guides/faq/faq.html>

Example Student Project

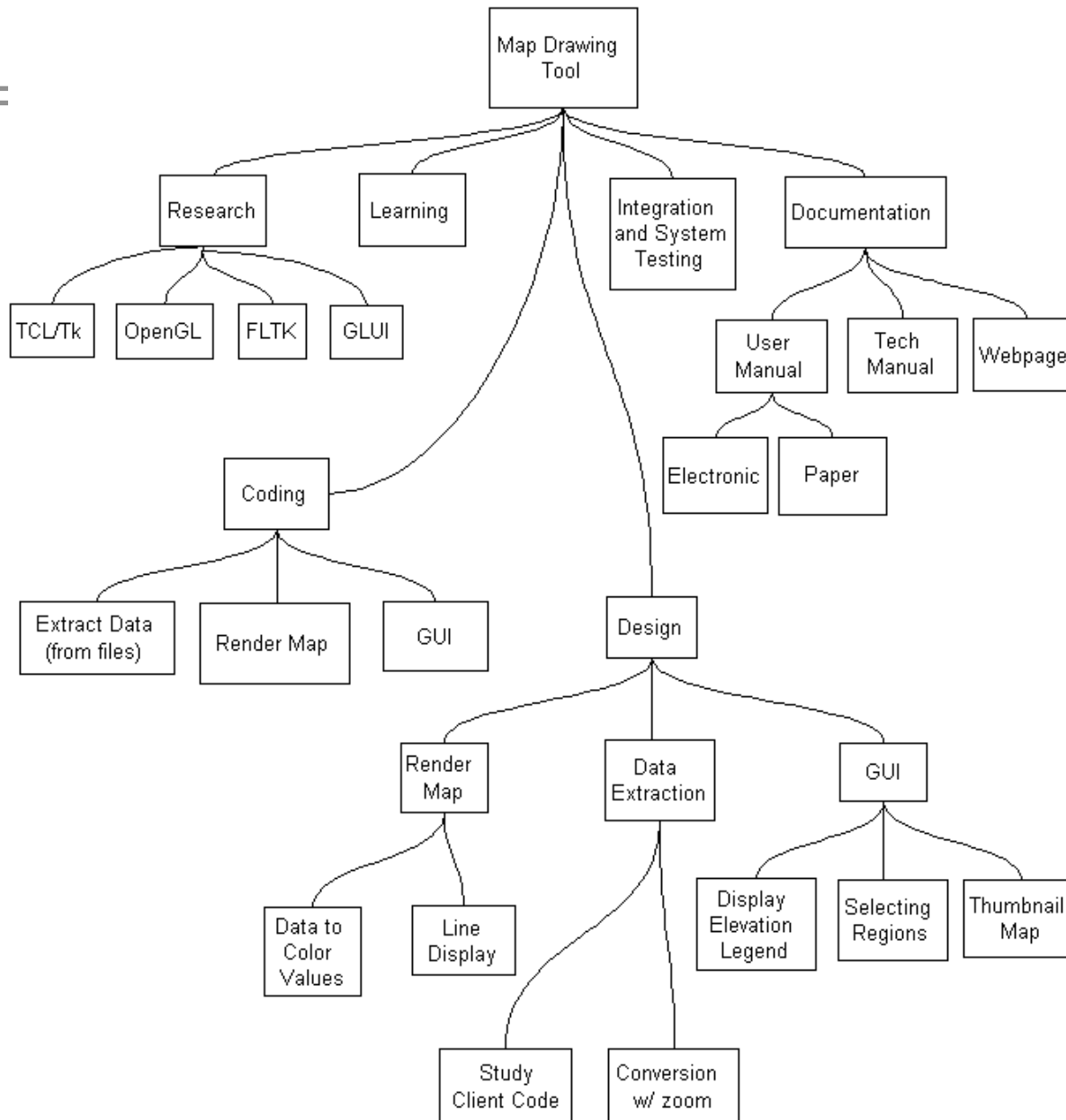
GISMO, or Geographic Information Systems MOdeller, is a map visualization tool designed to aid the user in viewing detailed terrain maps of a given area. GISMO presents an easy-to-use Graphical User Interface (GUI) for the viewing and manipulation of these maps. GISMO must be used in an environment that supports OpenGL. Within the Computer Science Department, only SGI machines support OpenGL, so GISMO has only been tested on these machines.

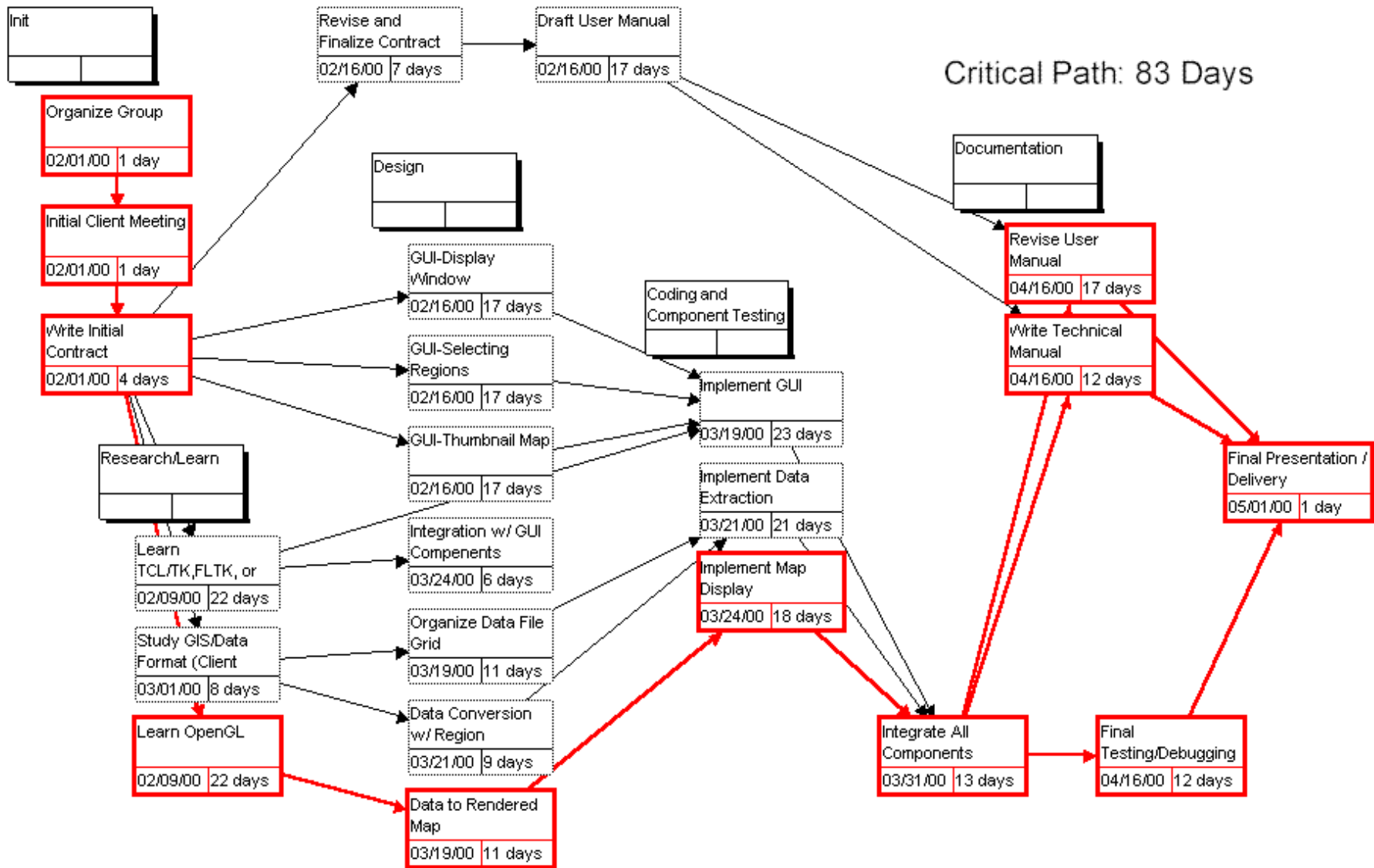
<http://www.cs.unc.edu/~stotts/COMP145/homes/map/description.html>

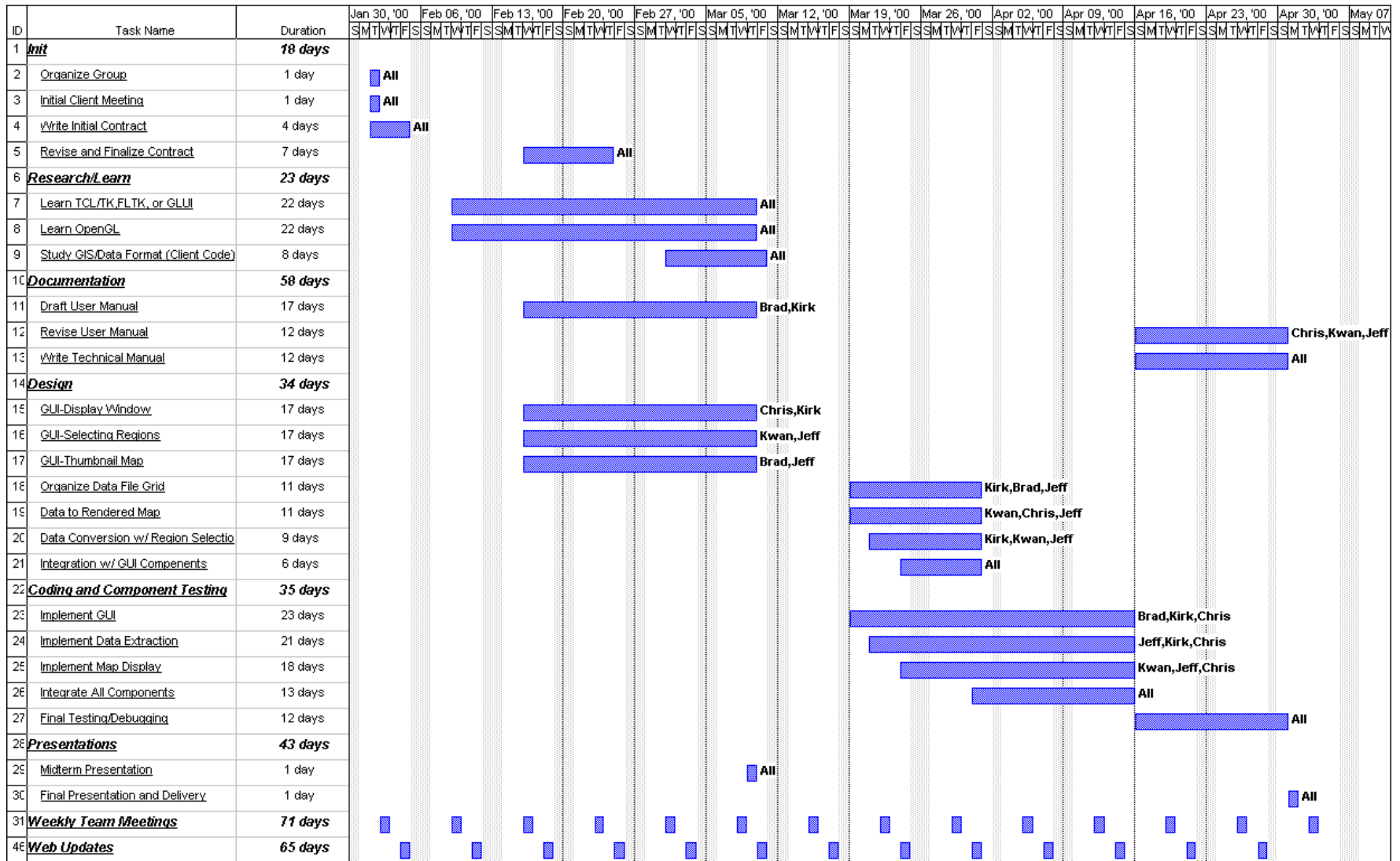
Milestones

<u>Task</u>	<u>Date Completed</u>	<u>Current Status</u>
Write and Sign Contract	February 24th	COMPLETED
Draft User Manual	March 9th	COMPLETED
Design - GUI	March 9th	COMPLETED
Design - Extract and Convert Data	March 31st	COMPLETED
Design - Render Map	March 31st	COMPLETED
All Components Implemented	April 15th	COMPLETED
Final Testing Completed	April 30th	COMPLETED
Write Technical Manual	April 30th	COMPLETED
Final Presentation / Delivery	May 4th	COMPLETED

Work Breakdown Structure







Control is the Goal

- Both are necessary for success!
- Intellectual control implies
 - We understand what we are trying to achieve
 - Can distinguish good choices from bad
 - We can reliably and predictably achieve what we want
- Managerial control implies
 - We make accurate estimations
 - We deliver on schedule and within budget
- Assertion: Managerial control is not really possible without intellectual control (no matter what the Harvard School of Business says)

Assignment

- Project