

Time Management II

<http://lbgeeks.com/gitc/pmTime.php>

June 5, 2008

- **Scheduling Methods**
- **Finding the Critical Path**
- **Scheduling Documentation**
- **Summary**

Scheduling Methods

- **Introduction**
- **Deadline Scheduling**
- **Priority Scheduling**
- **Sequence Scheduling**
- **Quantum Scheduling**

Scheduling Introduction

- **Parallel between project and real time operating system scheduling**
- **Some tasks must complete before a deadline:**
 - **Completing late causes failure**
 - **Various levels of failure severity**
- **Some tasks must be performed in a specific sequence**
- **Schedule other tasks for maximum throughput**

Deadline Scheduling

- **Task scheduled based on deadline**
 - **EDF: earliest deadline first**
 - **LSF: least slack first**
- **EDF produces feasible schedule**
- **Rate monotonic (RM) schedule:**
 - **Tasks have deadlines**
 - **Priority corresponds to frequency or slack**
 - **Higher priority tasks take less time**
- **RM produces optimal schedule**

Priority Scheduling

- **Goal: get most work done per day**
- **SJ(T)F: shortest job (task) first**
- **Print queue example: for jobs ready at same time, which should run first?**
- **Useful for personal time management:**
 - **Do unpleasant, important, quick tasks first**
 - **Rest of time devoted for project work**

Sequence Scheduling

- **Apply most resources at same time**
- **Identify constraints preventing activities (or tasks) from running in parallel:**
 - **Resource contention**
 - **Precedence order**
- **Partition or sequence tasks so people work independently without conflict**

Quantum Scheduling

- **Get most use of time-bounded resources**
- **Share among multiple consumers**
- **Each consumer receives a time slice called a quantum**
- **Consumer gets dedicated use of resource during quantum interval**
- **Used with other methods:**
 - **Consumer selection can be priority**
 - **Or can be simple sequence, round robin**

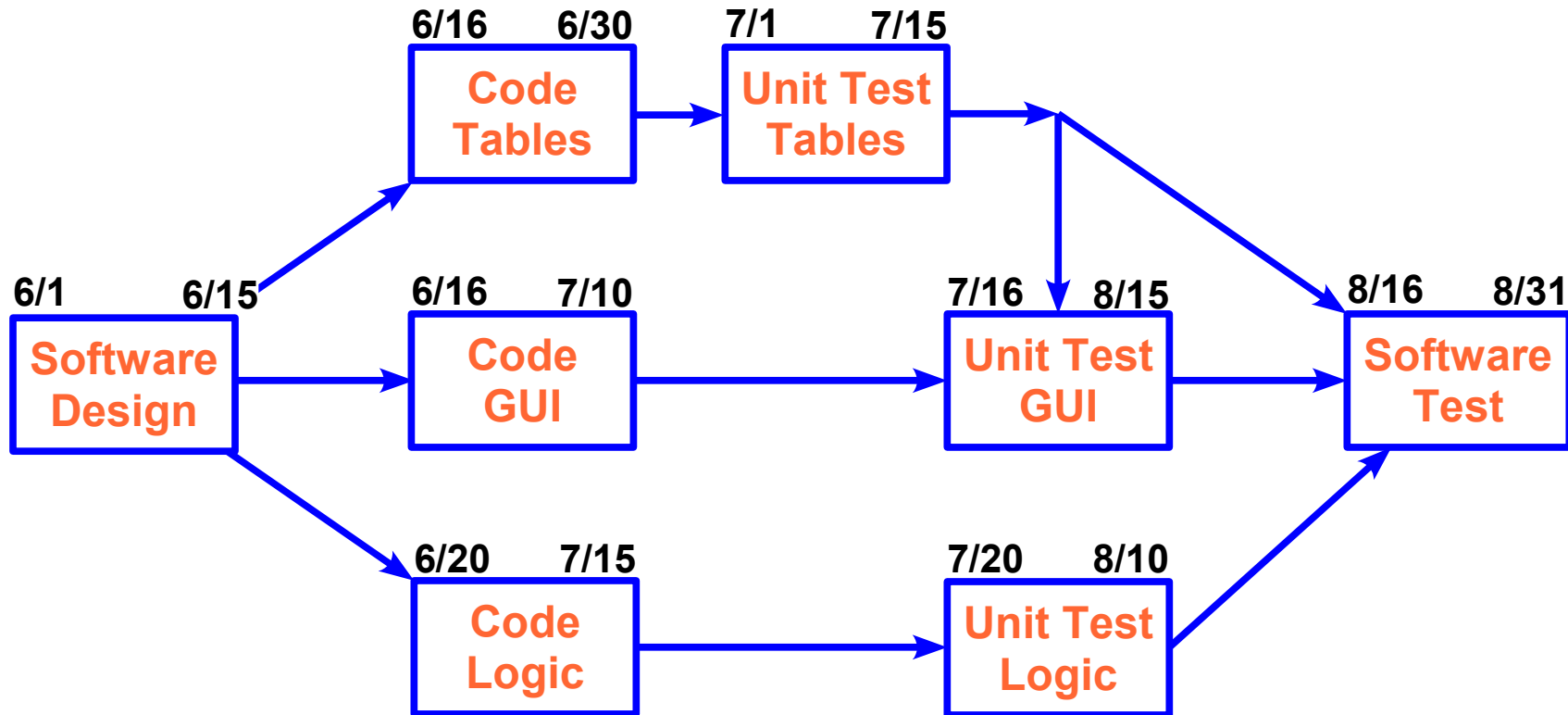
Finding the Critical Path

- **Definition**
- **Diagram**
- **Schedule Compression:**
 - **Crashing**
 - **Fast tracking**

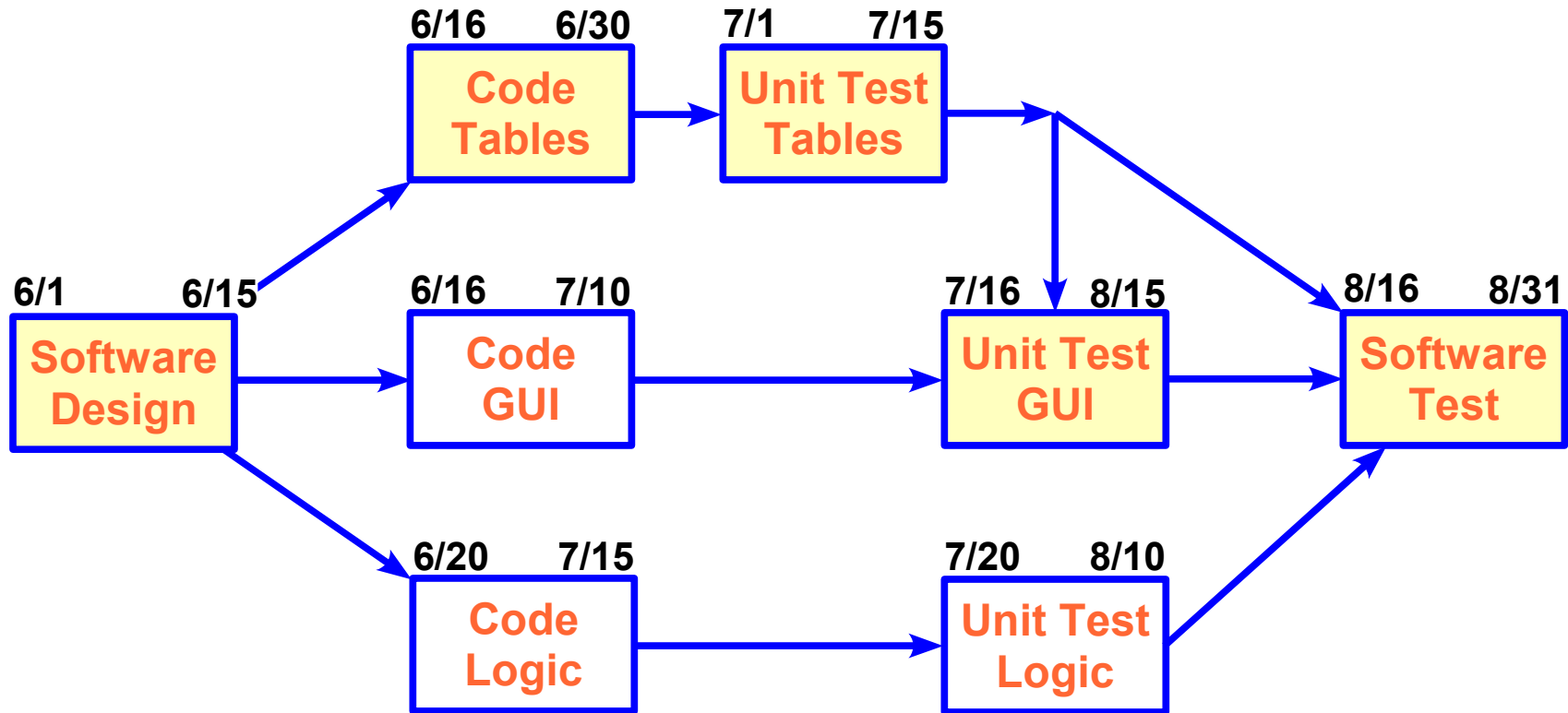
Critical Path Definition

- **Sequence of tasks where:**
 - **All must be completed exactly on time**
 - **If not, project will be delayed**
- **Output from at least one task is required input to another**
- **May be expressed as network flow or slack graph**

Critical Path Diagram – 1 of 2



Critical Path Diagram – 2 of 2



Schedule Compression

- **Crashing:**
 - **Consume more resources in less time**
 - **Sometimes produces more value**
 - **Example: lease additional computer**
- **Fast tracking:**
 - **Start dependent tasks before precedents fully complete**
 - **Fractional increase in parallelism**
 - **Difficult to manage**
 - **High risk**
 - **Example: start coding before design ends**

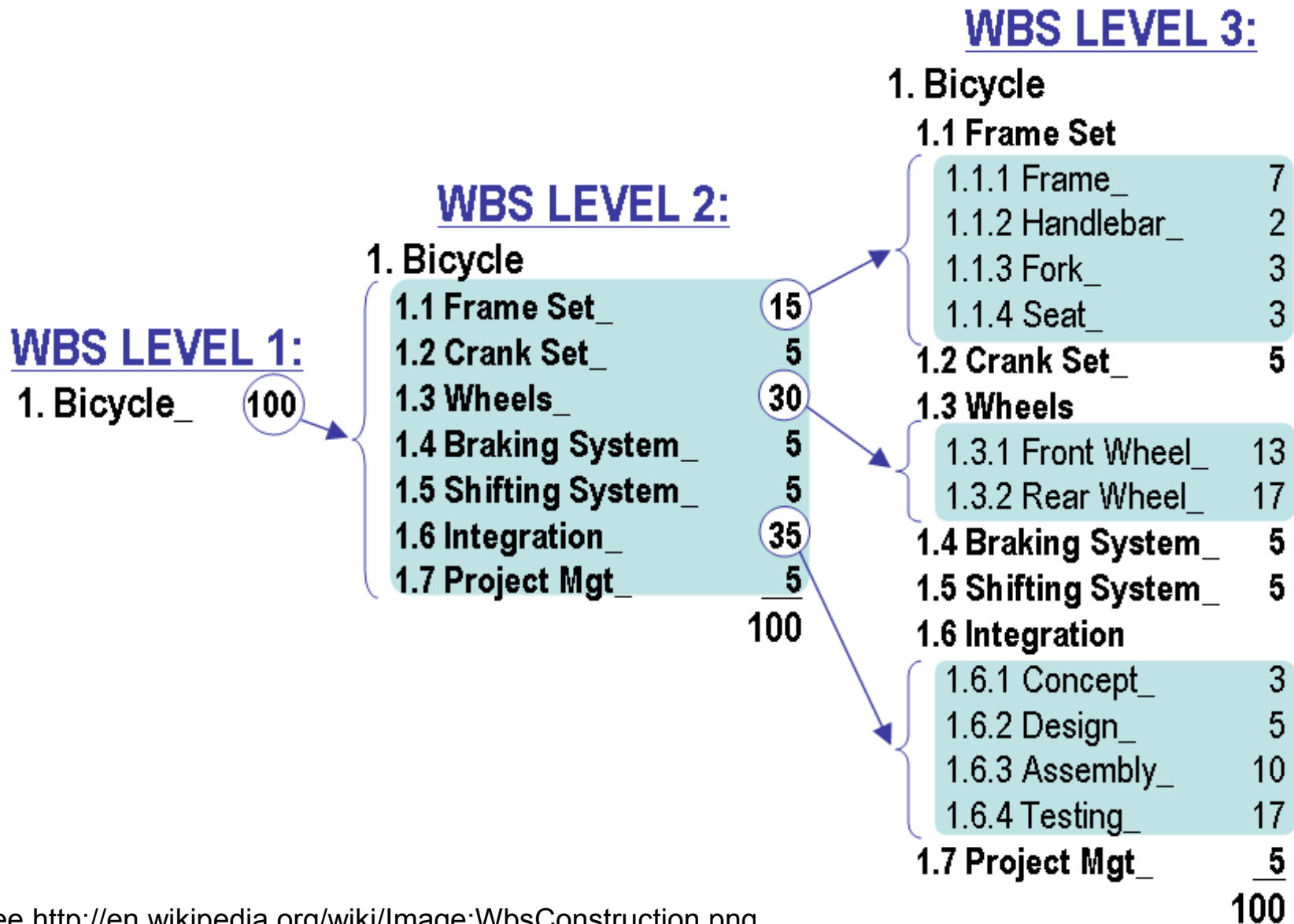
Scheduling Documentation

- **Work Breakdown Structure**
- **Gantt Charts**
- **Milestone Charts**
- **Time Scaled Diagrams**
- **Project Management Software**

Work Breakdown Structure

- **Defines work independent of time**
- **Tree structure showing allocation of effort to planned outcomes**
- **100% effort allocated to top level element**
- **No overlap between elements**
- **Terminal elements:**
 - **Atomic, not logically divisible, one person**
 - **Completed quickly, without interruption**
 - **Estimated completely, realistically**
- **Work packages group terminal elements**

Work Breakdown Structure Example

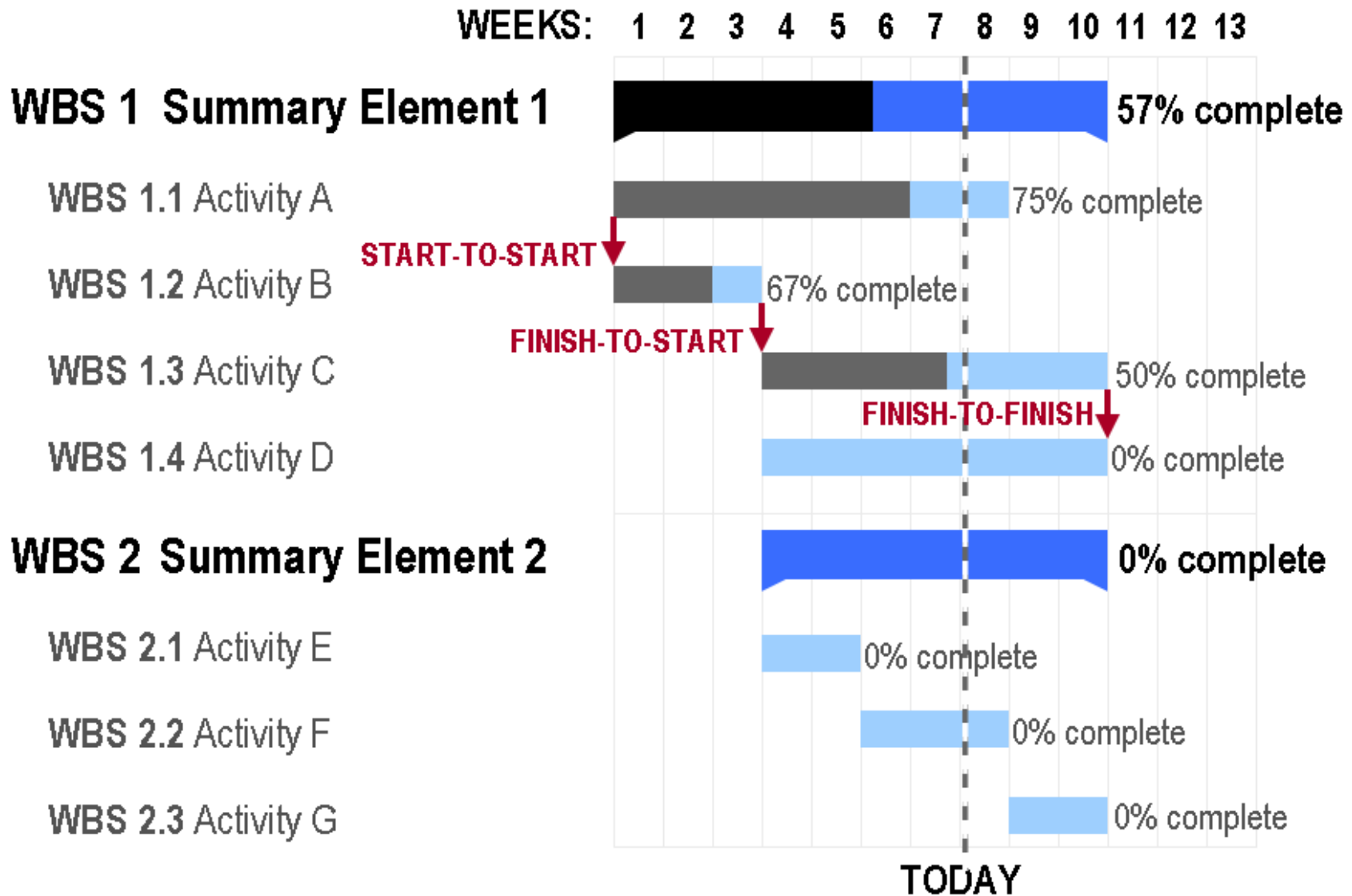


See <http://en.wikipedia.org/wiki/Image:WbsConstruction.png>

Gantt, Milestone Charts

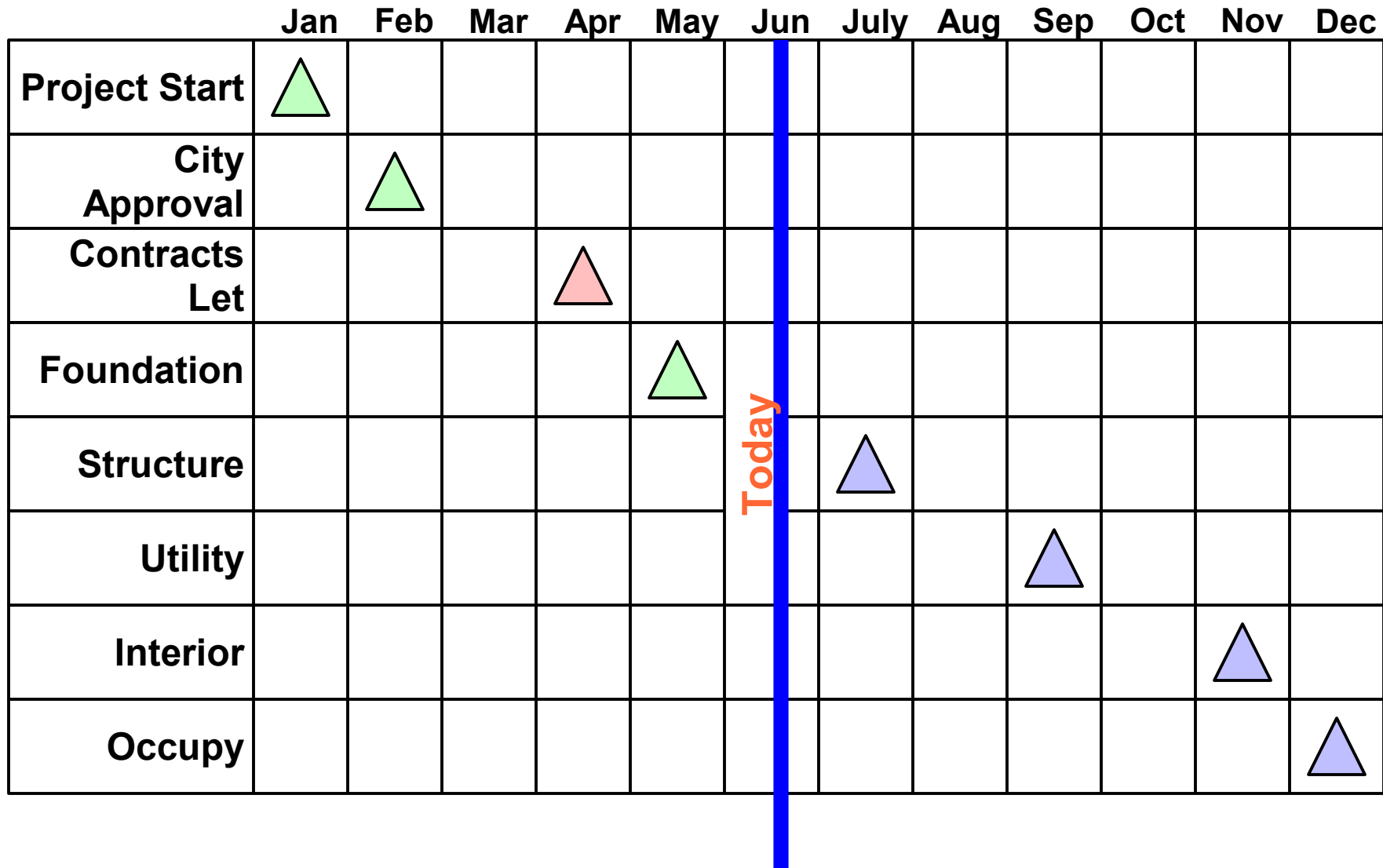
- **Gantt:**
 - Map time on work breakdown structure
 - Shows terminal elements in time order
 - May show network dependencies
 - Horizontal bar graph
 - X axis as time, Y axis as terminal elements
 - Percent complete and “today” line
- **Milestone:**
 - Same but shows time mapped on milestones
 - Used to document overall project state
 - Shows actual and estimated progress

Gantt Chart Example

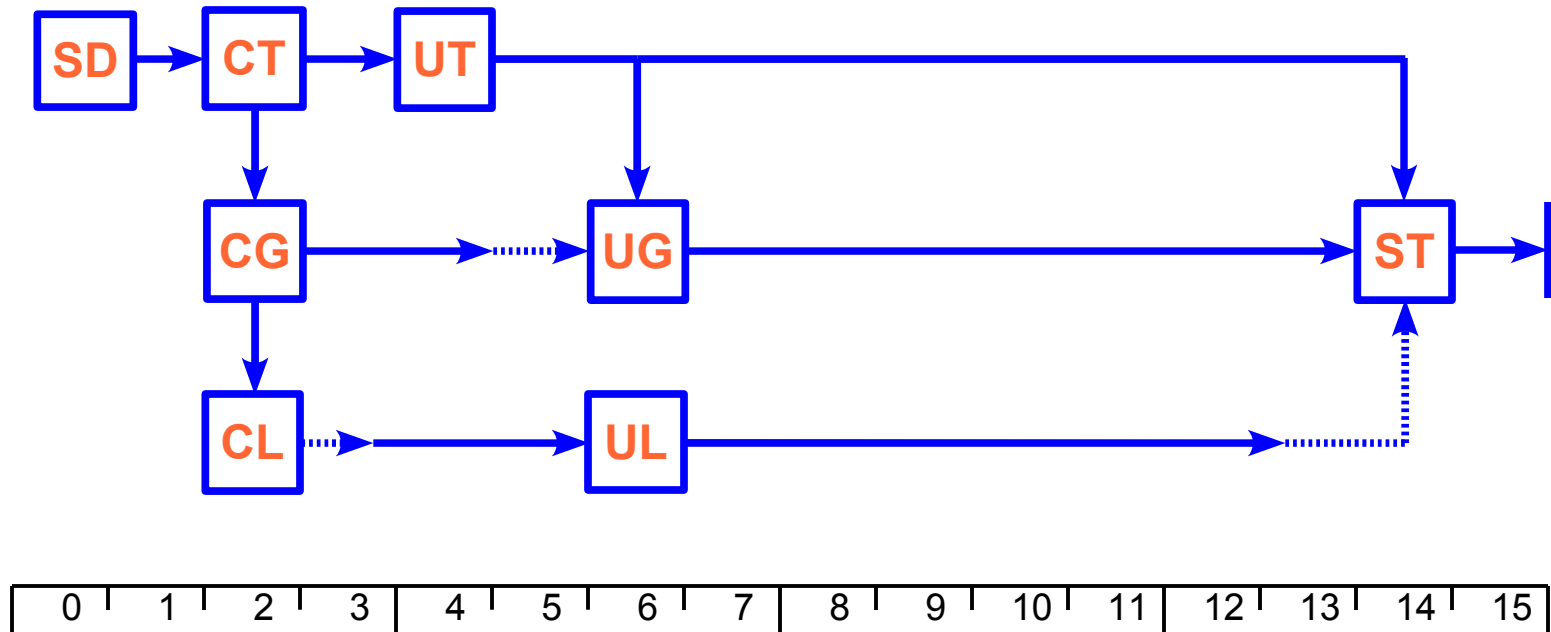


See <http://en.wikipedia.org/wiki/Image:GanttChartAnatomy.png>

Milestone Chart Example



Time Scaled Diagrams



SD = Software Design
CT = Code Tables
CG = Code GUI
CL = Code Logic

ST = Software Test
UT = Unit test Tables
UG = Unit test GUI
UL = Unit test Logic

Project Management Software

- **Projity Open Project**
 - **Open source MS Project clone**
 - **<http://www.openproj.org/product-overview>**
- **Gantt Project**
 - **Imports MS Project**
 - **Generates other charts from Gantt**
 - **<http://ganttproject.biz>**
- **We will use this software to complete course lab assignments**

Summary

- **Projects conduct activities using resources**
- **Estimates attempt to predict cost and time used during project**
- **Project time management requires basic understanding of graph theory**
- **Graphs can help us measure cost, time of multiple alternatives**
- **Specialized software produces project management specific documentation**