Project Management I

Software project management

Concerned with activities involved in ensuring that

- software is delivered on time
- and in accordance with the requirements of the organisation procuring the software.

Project management is needed because software development is always subject to

- budget
- and schedule constraints

that are set by the organisation procuring the software.

Software Management Challenges

The product is intangible.

The product is uniquely flexible.

Software engineering is not recognized as an engineering discipline with the same status as mechanical, electrical engineering, etc.

The software development process is not standardised.

Many software projects are 'one-off' projects.

"Software provides 'limitless scope' for functionality, without the inconvenient constraints of the laws of physics…There is nothing you can't do with software" (A. Bodmar)

From: The Challenges of Complex IT Projects, http://www.bcs.org/upload/pdf/complexity.pdf

Management activities

Proposal writing.

Project planning and scheduling.

Project costing.

Personnel selection and evaluation.

Project monitoring and reviews.

Report writing and presentations.

Software Project Management Plan

Software Project Management Plan:

- The controlling document for a software project.
- Specifies the technical and managerial approaches to develop the software product.
- The project plan sets out the resources available, the work breakdown, and the schedule
- Companion document to requirements specification: Changes in either may imply changes in the other document.
- SPMP may be part of the *project agreement*.

Project agreement: Document written for a client that defines:

- The scope, duration, cost and deliverables for the project.
- The exact items, quantities, delivery dates, delivery location.
- It can be a statement of work, a business plan, or a project charter – the basis for a *contract*

Software Project Planning

The project plan may address *aspects* of the project work, such as:

- **Quality plan**: quality procedures and standards to be used for system validation
- **Validation plan**: approach, resources and schedule for system validation
- **Configuration management plan**: configuration management procedures and structures used
- **Maintenance plan**: predicts maintenance requirements of system, maintenance costs and effort required
- **Staff development plan**: describes how the skills and experience of project team members are to be developed

And/or, aspects of the *development process*:

- **Introduction**: briefly, the objectives of project and sets constraints (e.g., budget, time, etc.) which affect project management
- **Project organisation**: way in which development team are organised, the people involved and their roles in the team
- **Risk analysis**: possible project risks, their likelihood of occurrence, and risk reduction strategies proposed
- **Hardware/software**: support required for the development; if hardware to be bought, estimates of price and delivery schedule
- **Work breakdown**: breakdown of project into activities and tasks, and identifies milestones and deliverables associated with each task
- **Project schedule**: dependencies between activities, estimated time required to reach each milestone, and allocation of people to activities
- **Monitoring and reporting mechanisms**: management reports which should be produced, when, and the monitoring mechanism.

Road Map for Project Management

1. Understand the project content, scope & time frame
2. Identify development process
   - methods, tools, languages, documentation and support
3. Determine organizational structure
   - organizational elements involved
4. Identify managerial process
   - responsibilities of the participants
5. Develop schedule
   - times at which the work portions are to be performed
6. Develop staffing plan
7. Begin risk management
8. Identify documents to be produced
9. Begin the process...
   - On-going monitoring, reviewing...

Project scheduling

Split project into tasks and estimate time and resources required to complete each task.

Organize tasks concurrently to make optimal use of workforce.

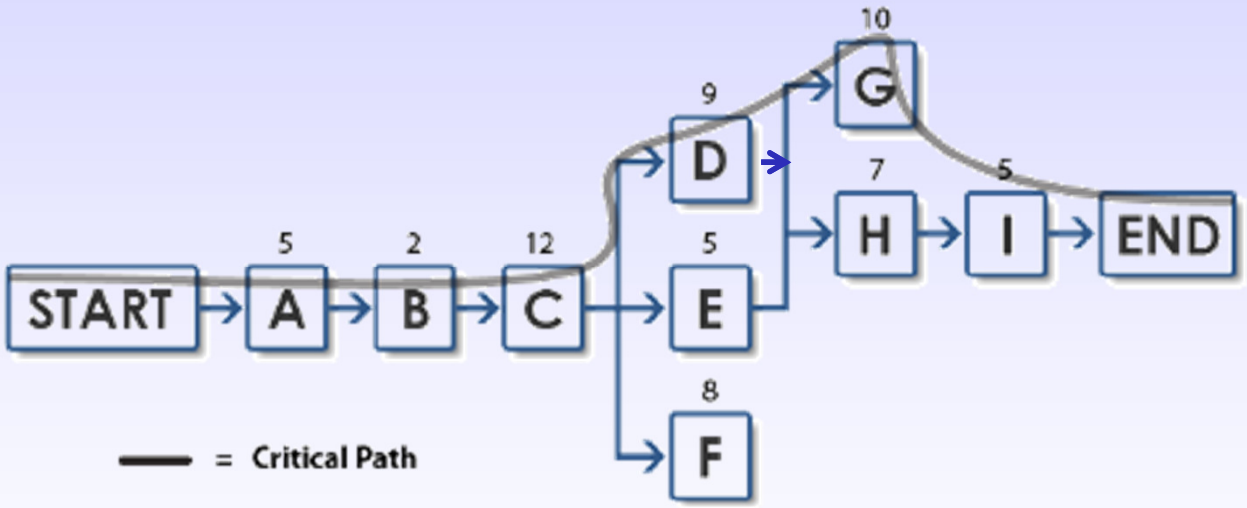
Minimize task dependencies to avoid delays caused by one task waiting for another to complete.

Dependent on project manager's intuition and experience.

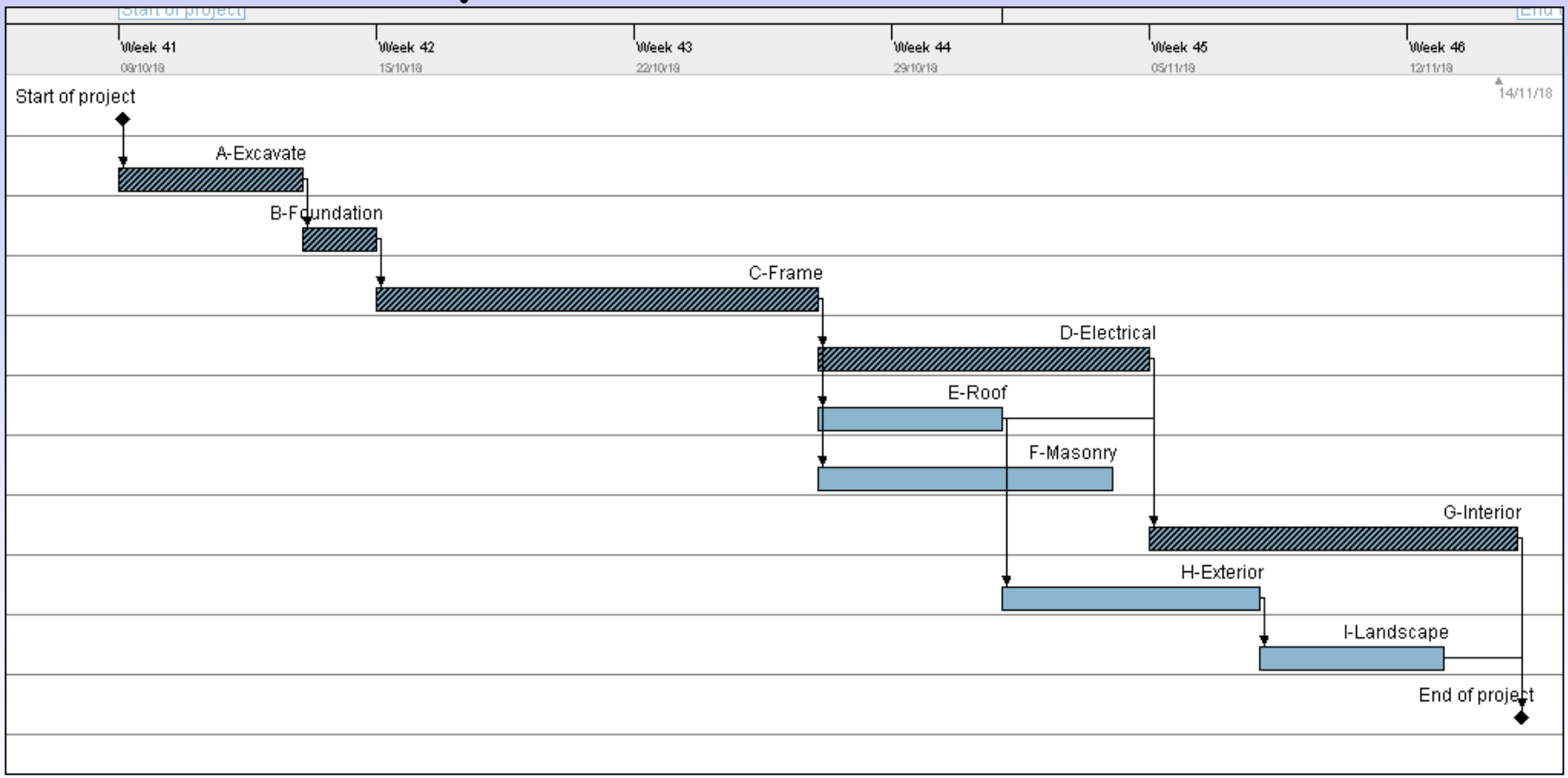
Activity network

Example: activity plan for building a house. (Taken from www.sigsigmadaily.com/the-activity-network-diagram)

A. Excavate	5 days
B. Foundation	2 days
C. Frame	12 days
D. Electrical	9 days
E. Roof	5 days
F. Masonry	8 days
G. Interior	10 days
H. Exterior	7 days
I. Landscape	5 days



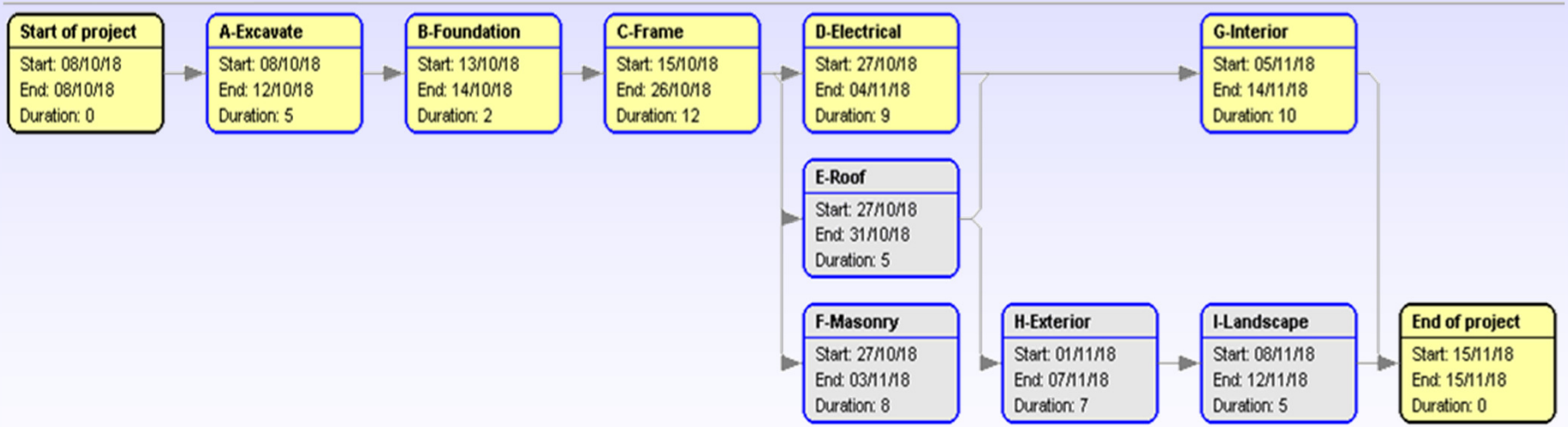
Activity timeline: Gantt Chart



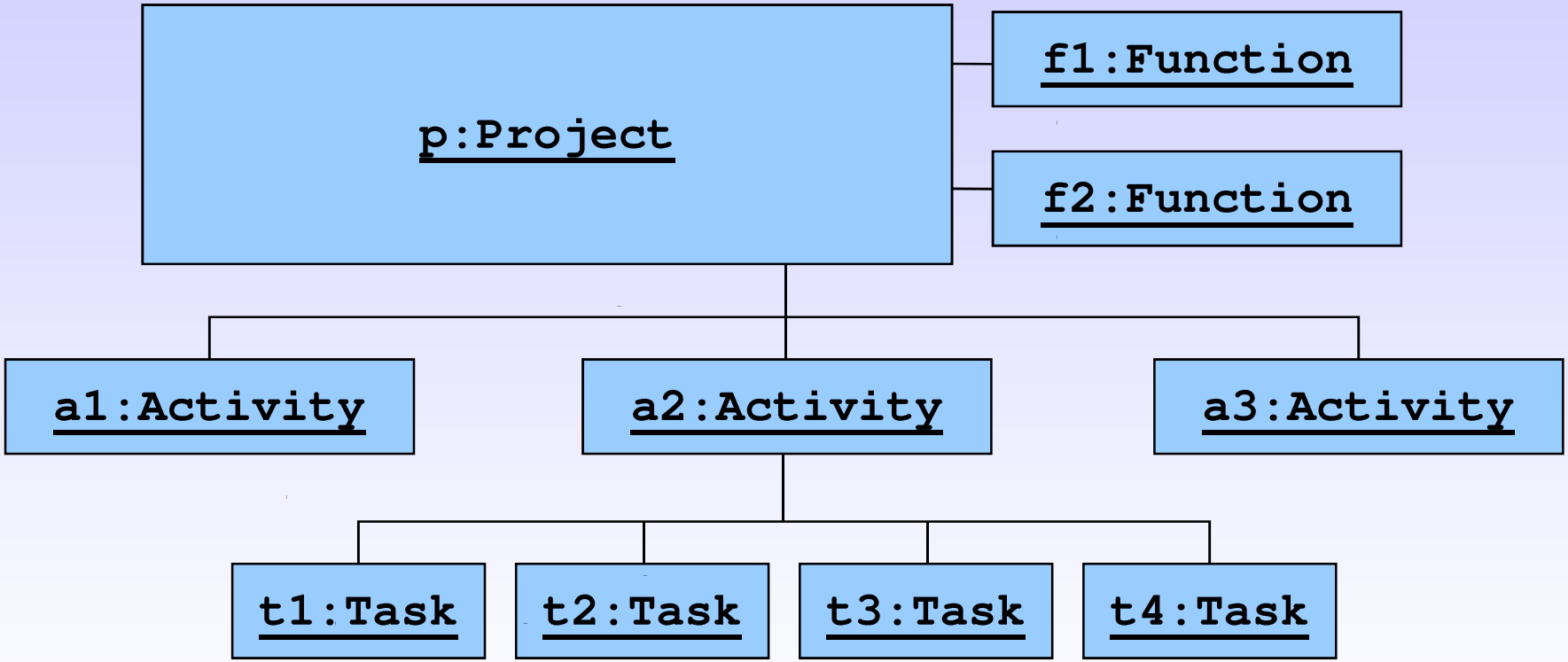
- Named after Henry Gantt 1861-1919, engineer & management consultant
- Gantt charts can also show resources, other dependencies, and critical paths
- In the labs: GanttProject: see www.ganttproject.biz)

PERT chart

- "Program Evaluation and Review Technique"
- Developed for the US Navy in the 1950s.
- Similar to activity diagram, but can more include detailed timing information.



Functions, Activities & Tasks



Functions

Activity or set of activities that span the duration of the project

Examples:

- Project management
- Configuration Management
- Documentation
- Quality Control (Verification & Validation)
- Training

Activities

Major unit of work with precise dates

Consists of smaller activities or tasks

Culminates in major ***project milestone***:

- Internal checkpoint that should not be externally visible
- Scheduled event used to measure progress

Milestone often produces ***baseline***:

- Formally reviewed work product
- Under change control (change requires formal procedures)
- A fixed point to work on from

Activities may be grouped into larger activities:

- Establishes hierarchical structure for project
- Allows separation of concerns
- Precedence relations often exist among activities

Examples of Activities

Major Activities

- Planning
- Requirements Elicitation
- Requirements Analysis
- System Design
- Object Design
- Implementation
- System Testing
- Delivery

Activities during Requirements Analysis:

- Refine scenarios
- Define Use Case model
- Define object model
- Define dynamic model
- Design User Interface

Tasks

Smallest unit of management accountability

Small enough for adequate planning and tracking

Large enough to avoid micromanagement

- Atomic unit of planning and tracking
- Finite duration, need resources, produce tangible result (documents, code)

Specification of a task: A Work Package

- Name, description of work to be done
- Preconditions for starting, duration, required resources
- Work product to be produced, acceptance criteria for it
- Risk involved

Completion criteria

- Includes the acceptance criteria for the work products (deliverables) produced by the task.

Task Sizes

Finding the appropriate task size is problematic

- During initial planning a task is necessarily large
- You may not know how to decompose the problem into tasks at first
- Each software development activity identifies more tasks and modifies existing ones

Tasks must be decomposed into sizes that allow monitoring

- Work package usually corresponds to well defined work assignment for one worker for a week or a month.
- Depends on nature of work and how well task is understood.
- Could look at "to do" lists from previous projects

Examples of Tasks

- Unit test class "Foo"
- Test subsystem "Bla"
- Write user manual
- Write meeting minutes and post them
- Schedule the code review
- Action item

Action Item

Definition: A very clearly delimited task assigned to a person that has to be done within, say, a week or less

Action Items

- Covers: What?, Who?, When?

Example of *Action Items*

- Jim unit tests class "Foo" by next week
- Sue develops a project plan before the next meeting
- Bob posts the next agenda for the Simulation team meeting before Sep 10, 12noon.

End of lecture