



# Lecture 11: Managing Your Project

## The task of Management

### Planning the project

Task breakdown  
Pert Charts, Gantt Charts

### Measuring Progress

Burndown charts  
Test Progress Charts

### Use of meetings



# Starting point

## You have a prioritized list of tasks

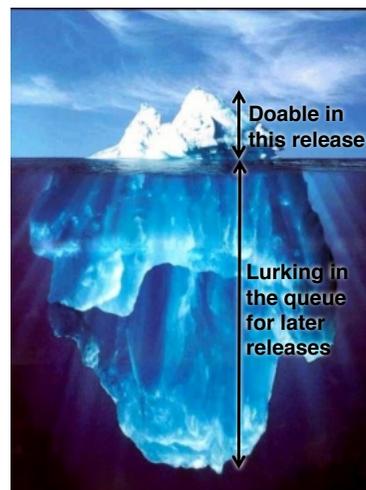
...entered into the tracking system  
...each task has an effort estimate

## You have selected which tasks go into the release

“Iceberg list”: top n items in this release  
additional requests from customer push other items below the waterline  
faster than expected progress lifts items up above the waterline

## Problem:

How do we keep the project on track?





## Management Duties

- Analyzing and Controlling Risk
- Liaising with customer
- Defining lines of communication
- Recruiting and training project team
- Create project plan, schedule, cost estimate,...
- Assign tasks
- Measure progress of project
- Make sure appropriate tools and techniques are used
- Keep project on track, take action if it slips
- Ensure contractual obligations met
- Ensure appropriate standards are met
- Ensure lessons learned are captured and used



## Refresher: project control

### A manager can control 4 things:

- Resources** (can get more dollars, facilities, personnel)
- Time** (can increase schedule, delay milestones, etc.)
- Product** (can reduce functionality - e.g. scrub requirements)
- Risk** (can decide which risks are acceptable)

### To do this, a manager needs to keep track of:

- Effort** - How much effort will be needed? How much has been expended?
- Time** - What is the expected schedule? How far are we deviating from it?
- Size** - How big is the planned system? How much have we built?
- Defects** - How many errors are we making? How many are we detecting?  
And how do these errors impact quality?

### Initially, a manager needs good estimates

...and these can only come from a thorough analysis of the problem.

**You cannot control that which you cannot measure!**





# Tool 1: Work Breakdown Structure

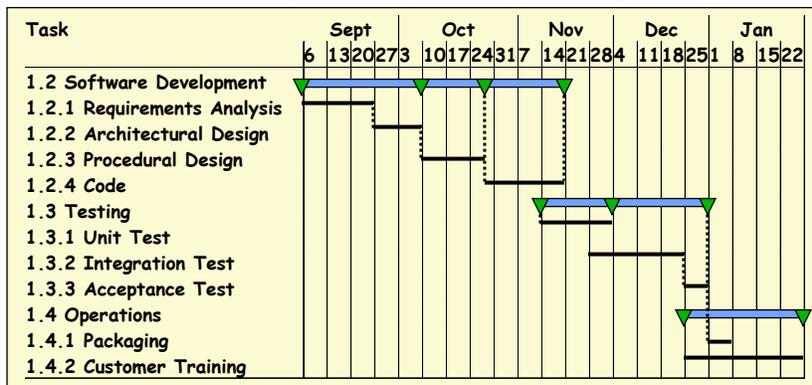
Source: Adapted from Blum, 1992, p438  
see also: van Vliet pp192-3

- 1.1 Software Systems Engineering**
  - 1.1.1 Support to Systems Engineering
  - 1.1.2 Support to Hardware Engineering
  - 1.1.3 Software Engineering Trade Studies
  - 1.1.4 System Requirements Analysis
  - 1.1.5 Software Requirements Analysis
  - 1.1.6 Interface Analysis
  - 1.1.7 Support to Systems Test
- 1.2 Software Development**
  - 1.2.1 Deliverable Software
    - 1.2.1.1 Requirements Analysis
    - 1.2.1.2 Architectural Design
      - 1.2.1.3 Procedural Design
      - 1.2.1.4 Code
      - 1.2.1.5 Unit Test
      - 1.2.1.6 Software Integration Test
      - 1.2.1.7 Technical Reviews
      - 1.2.1.8 Technical Training
  - 1.2.2 Non-deliverable Software
  - 1.2.3 Purchased Software
    - 1.2.3.1 Package Evaluation
  - 1.2.4 Development facilities and tools
- 1.3 Software Test and Evaluation**
  - 1.3.1 Software Dev. Test & Evaluation
  - 1.3.2 End-Product Acceptance Test
  - 1.3.3 Test Bed & Tool Support
  - 1.3.4 Test Data Management
- 1.4 Management**
  - 1.4.1 Project Management
  - 1.4.2 Administrative Support
  - 1.4.3 Management Tools
  - 1.4.4 Management Reviews
  - 1.4.5 Management Training
- 1.5 Product Assurance**
  - 1.5.1 Configuration Management
  - 1.5.2 Library Operations
  - 1.5.3 Interface Control
  - 1.5.4 Data Management
  - 1.5.5 Quality Assurance
  - 1.5.6 Quality Control
- 1.6 Operations and Support**
  - ...



see also: van Vliet pp195-6

# Tool 2: Gantt Charts



## Notation

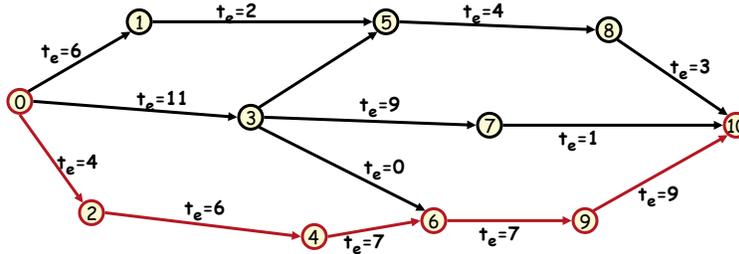
- Bars show duration of tasks
- Triangles show milestones
- Vertical dashed lines show dependencies

Shows high level view of whole project



# Tool 3: PERT charts

Source: Adapted from Blum, 1992, p439  
see also: van Vliet pp193-6



## → Notation

- ↳ Nodes indicate milestones
- ↳ Edges indicate dependencies
- ↳ Edges are labelled with time to complete

## → Shows Critical Path

- ↳ Longest path from start to finish
- ↳ any slippage on the critical path will cause project delay



# Avoiding Gantt Charts?

Source: Adapted from Rosenberg's ICONIX book, p182

## Gantt charts can be misleading

- “% complete” for most tasks is valueless
- poorly defined completeness criteria for each task

## High level management view:

- Is the project on track?
- Is the next release going to be late?
- What will be in the next release?
- Has anything slipped to the following release?

## Risk-based planning:

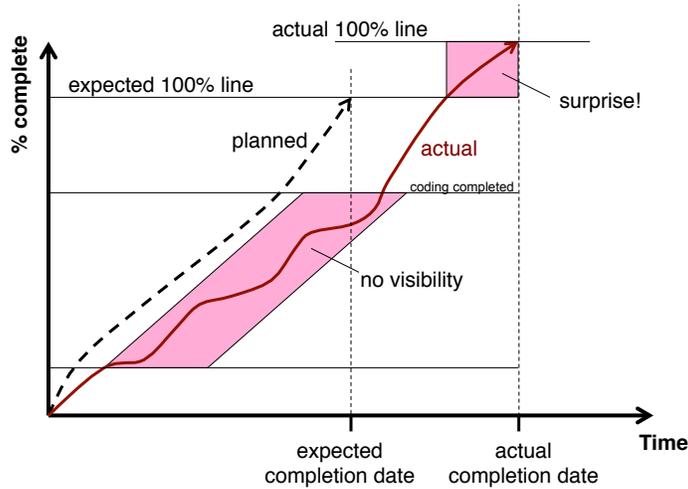
- What can go wrong?
- What are the consequences of not knowing that they are going wrong?





# Measuring Progress

Source: Adapted from Cockburn, "Crystal Clear"



# Agile Approach

## Measure progress of small tasks

- Reduces regions of no visibility
- Use the issue tracking system!

## Estimate effort for each task

- adjust estimate of remaining effort as iteration progresses

## Only working code counts

- Task complete only when it delivers tested, working code
- (okay, training material, manuals, etc also count - but nothing else!)

## Visualize Progress

- E.g. use a burndown chart

## Use your experience!

- E.g. if estimates are wrong on this iteration, adjust estimates for the next



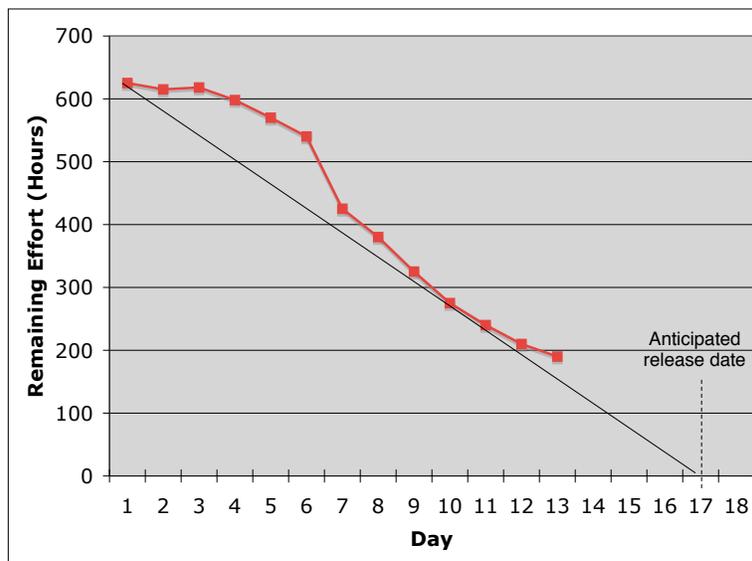
# Maintain a task list

Source: Adapted from Cockburn, "Crystal Clear"

Module	Feature Name	Value	Raw Dev. Time (ideal days)	Estimated Elapsed Days	release #
1.1	Configuration generates order line item comments	M	2	5	1
1.2	Configurator UI Rework: Verbose wizard style	H	6	15	1
2.1	PO generated correctly for configuration	H	5	12.5	1
2.2	Create/Confirm vendor items exist for skus	H	1.5	3.75	1
3.1	Advanced Order form shows more details	M	4	10	1
3.2	Order fulfilled at PO cost	H	2	5	1
3.3	Repeat orders works with blind configurations	M	3	7.5	1
3.4	Configuration comments are viewable, not editable	L	1	2.5	1
4.1	Base hierarchy change	H	1.5	3.75	1
4.2	Style can locate price charts based on color selection	H	2	5	1
4.3	Assign specific color group colors to price charts	H	2	5	1
4.4	Separate messages from options	H	2	5	1
4.5	Separate questions from options	H	2	5	1
1.3	Configuration generates customer specific pricing	H	5	12.5	2
1.4	Allow adding a configuration and continue	L	2	5	2
2.3	PO generation is automatic	L	4	10	2

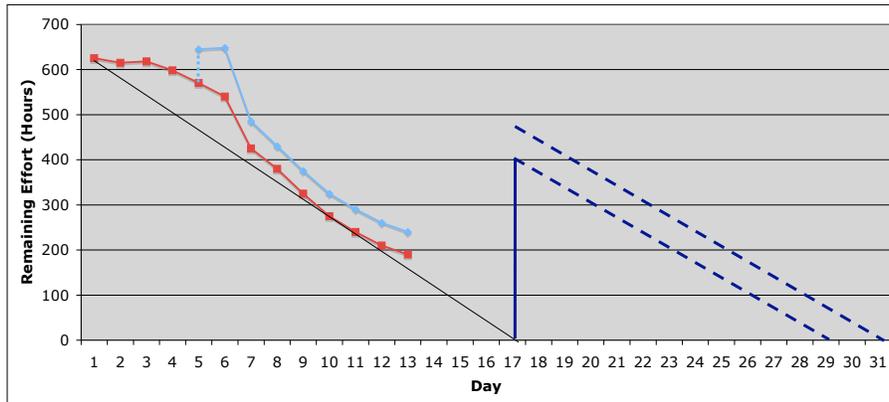


# Burn-down Chart

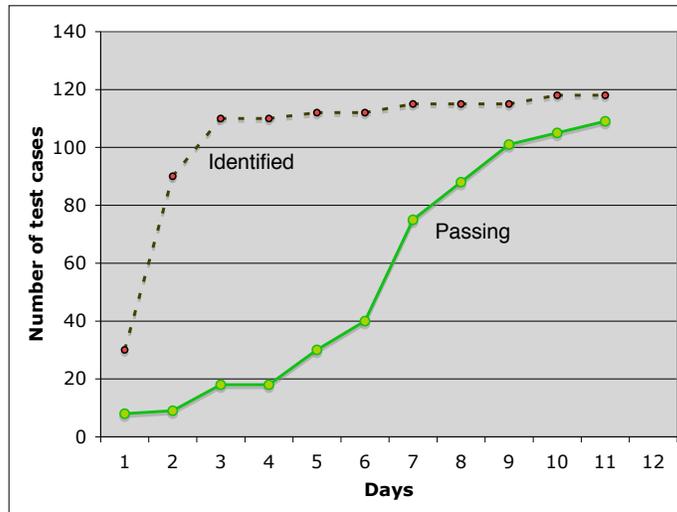




### Multi-release trade-offs



### Test Case Trend





## Meetings as a management tool

Source: Adapted from Pfeleger, 1998, 92

### Meetings are expensive

E.g. 8 people on \$60k. Meeting costs \$480 per hour

### Meetings are necessary

Can save money by averting misunderstandings and coordination errors

### Time wasters:

Purpose of meeting unclear  
Attendees unprepared  
Essential people missing  
Discussion gets sidetracked  
Dominance by one or two people  
Argumentative  
Decisions not followed up on

### Meetings advice:

- ☞ **Announce details in advance**
  - > who should attend
  - > start and end times
  - > goals of meeting
- ☞ **Clear agenda, distributed in advance**
- ☞ **Identify a chairperson who:**
  - > keeps the discussion on track
  - > resolves arguments
- ☞ **Identify a secretary who:**
  - > keeps track of decisions taken
  - > records action items
  - > ensures action items are carried out
- ☞ **Associate a responsible person with each action item**



## Discussion Question

**Management**  
**=**  
**Leadership**  
**??**

