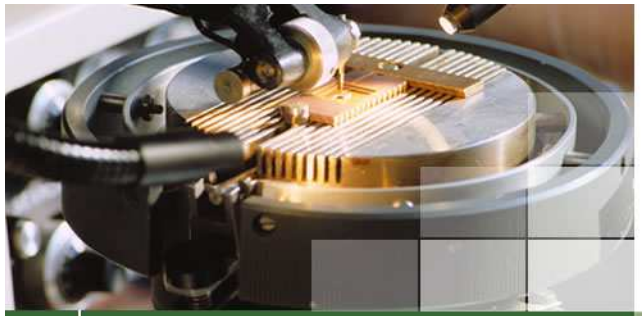


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Planning and Managing Software Projects 2010-11

Session 4

WBS, and Estimation

Emanuele Della Valle

http://emanueledellavalle.org

Credits

- This slides are largely based on Prof. John Musser class notes on “Principles of Software Project Management”
- Original slides are available at http://www.projectreference.com/
- Reuse and republish permission was granted

Today & *Next Week*

- Review Session 3
- Work Breakdown Structures (WBS)
- Estimation

Session 3 Review

- Full Lifecycle
  - Know your pure waterfall, 7 phase model
  - Understand the steps in each phase
  - Know the primary issues and goals of each
- Methodologies
  - Know a representative sample
  - Waterfall and variation, 1-2 iterative ones
  - Learn a bit about XP and other Agile methods
- Planning (introduction)
  - Primary Planning Steps
  - Documents

Estimation

- “Predictions are hard, especially about the future”

  - Yogi Berra*

- 2 Types: Lucky or Lousy?

* http://en.wikipedia.org/wiki/Yogi_Berra

Planning, Estimating, Scheduling

- What's the difference?
- Plan: Identify activities. No specific start and end dates.
- Estimating: Determining the size & duration of activities.
- Schedule: Adds specific start and end dates, relationships, and resources.

Project Planning: A 12 Step Program

1. Set goal and scope
2. Select lifecycle
3. Set team form
4. Start team selection
5. Determine risks
6. Create WBS
7. Identify tasks
8. Estimate size
9. Estimate effort
10. Identify task dependencies
11. Assign resources
12. Schedule work

How To Schedule

1. Identify “what” needs to be done
   - Work Breakdown Structure (WBS)
2. Identify “how much” (the size)
   - Size estimation techniques
3. Identify the dependency between tasks
   - Dependency graph, network diagram
4. Estimate total duration of the work to be done
   - The actual schedule

WBS & Estimation

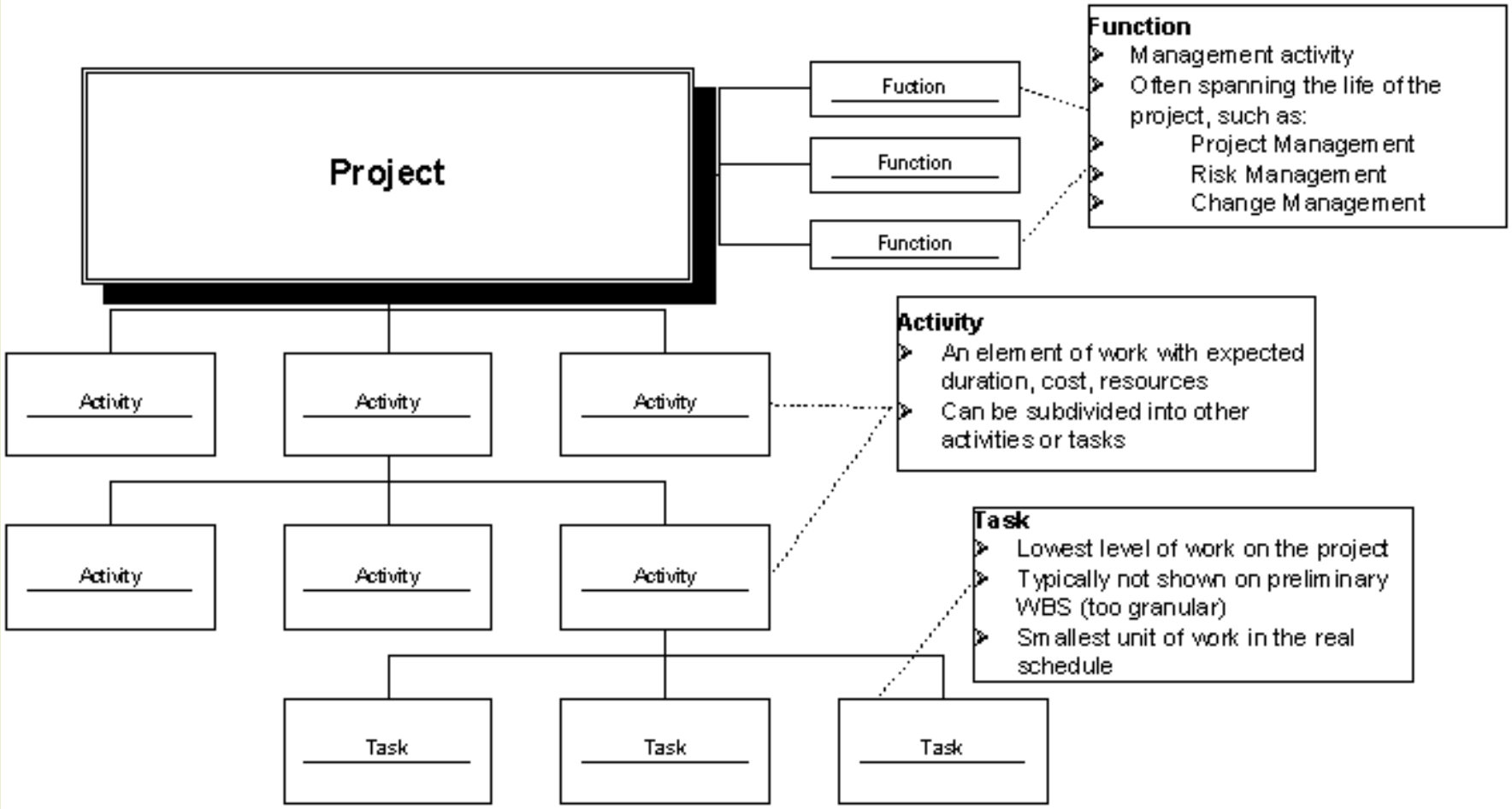
- How did you feel when I asked
  - “How long will your project take?”
- Not an easy answer to give right?
- At least not if I were are real customer on a real project
- How can you manage that issue?

Partitioning Your Project

- You need to decompose your project into manageable chunks
- ALL projects need this step
- Divide & Conquer
- Two main causes of project failure
  - Forgetting something critical
  - Ballpark estimates become targets
- How does partitioning help this?

Project Elements

- A Project: functions, activities, tasks



Work Breakdown Structure: WBS

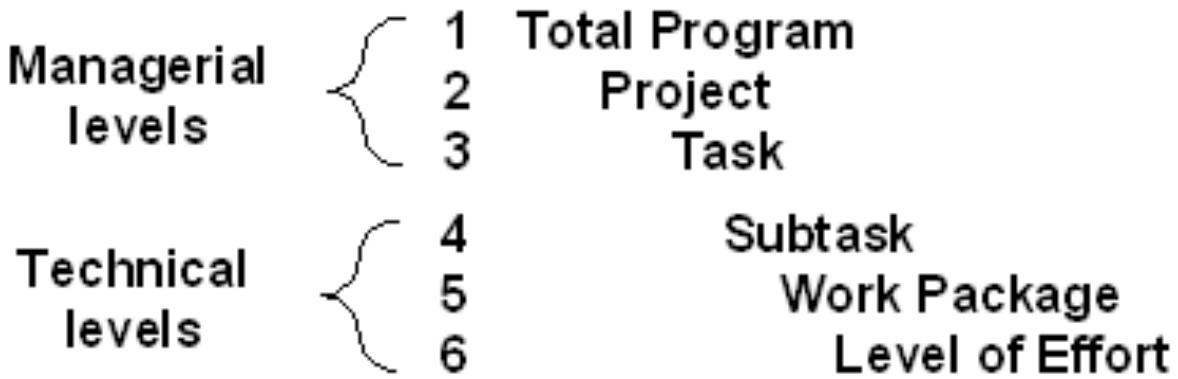
- Hierarchical list of project's work activities
- 2 Formats
  - Outline (indented format)
  - Graphical Tree (Organizational Chart)
- Uses a decimal numbering system
  - Ex: 3.1.5
- Includes
  - Development, Mgmt., and project support tasks
- Shows "is contained in" relationships
- Does not show dependencies or durations

Contract vs. Project WBS

- Contract WBS (CWBS)
  - First 2 or 3 levels
  - High-level tracking
- Project WBS (PWBS)
  - Defined by PM and team members
  - Tasks tied to deliverables
  - Lowest level tracking

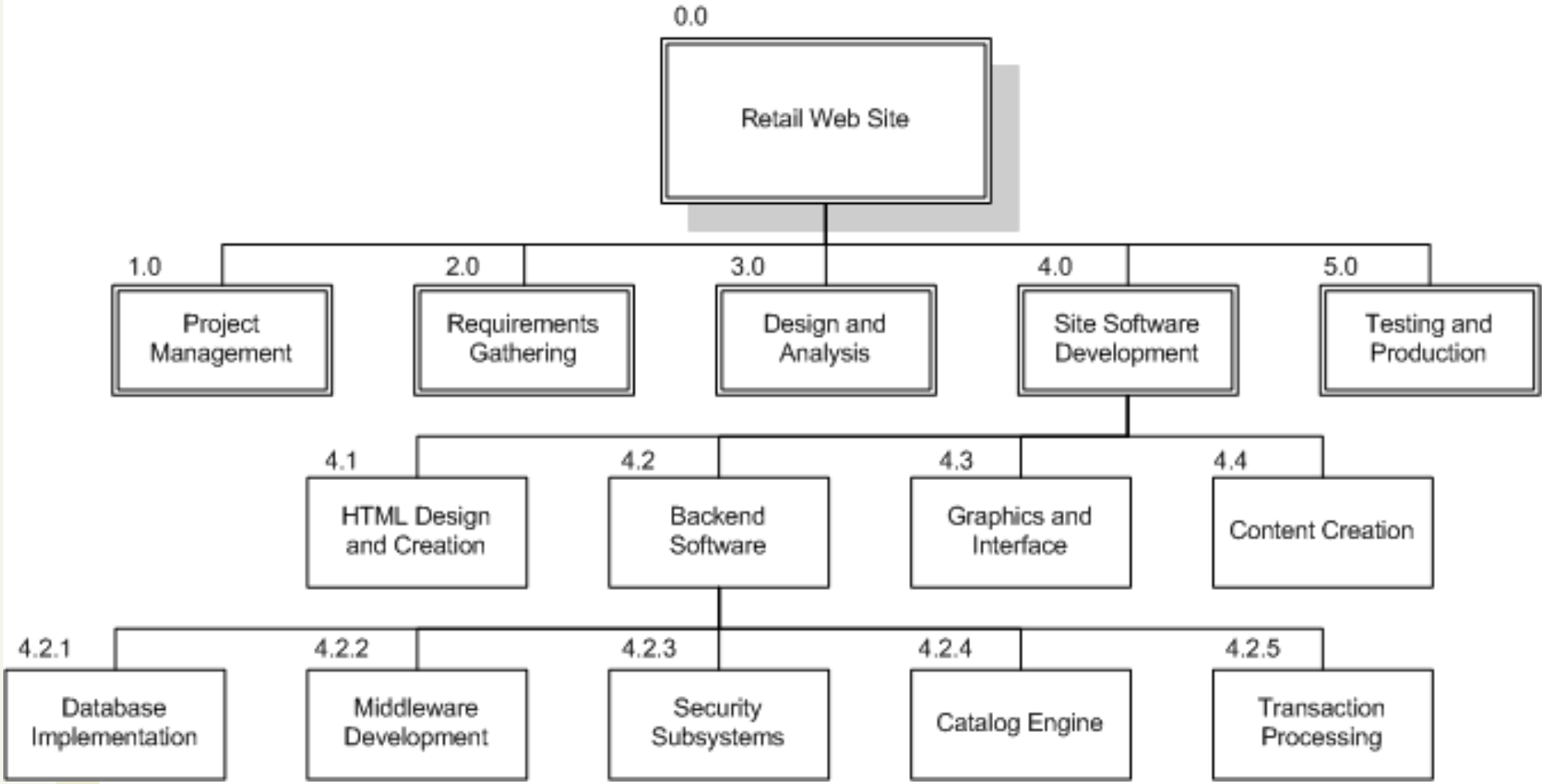
A Full WBS Structure

- Up to six levels (3-6 usually) such as



- Upper 3 can be used by customer for reporting (if part of RFP)
- Different level can be applied to different uses
  - Ex:
    - Level 1: authorizations
    - Level 2: budgets
    - Level 3: schedules

Chart Example



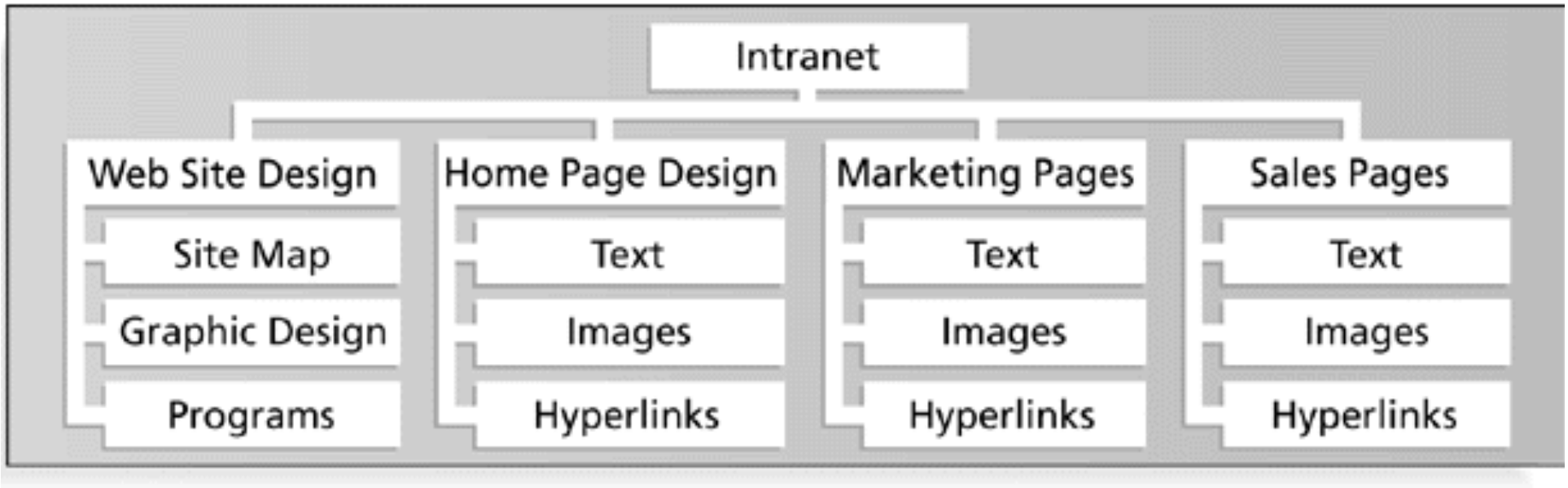
Outline Example

Retail Web Site
1.0 Project Management
2.0 Requirements Gathering
3.0 Analysis & Design
4.0 Site Software Development
    4.1 HTML Design and Creation
    4.2 Backend Software
        4.2.1 Database Implementation
        4.2.2 Middleware Development
        4.2.3 Security Subsystems
        4.2.4 Catalog Engine
        4.2.5 Transaction Processing
    4.3 Graphics and Interface
    4.4 Content Creation
5.0 Testing and Production

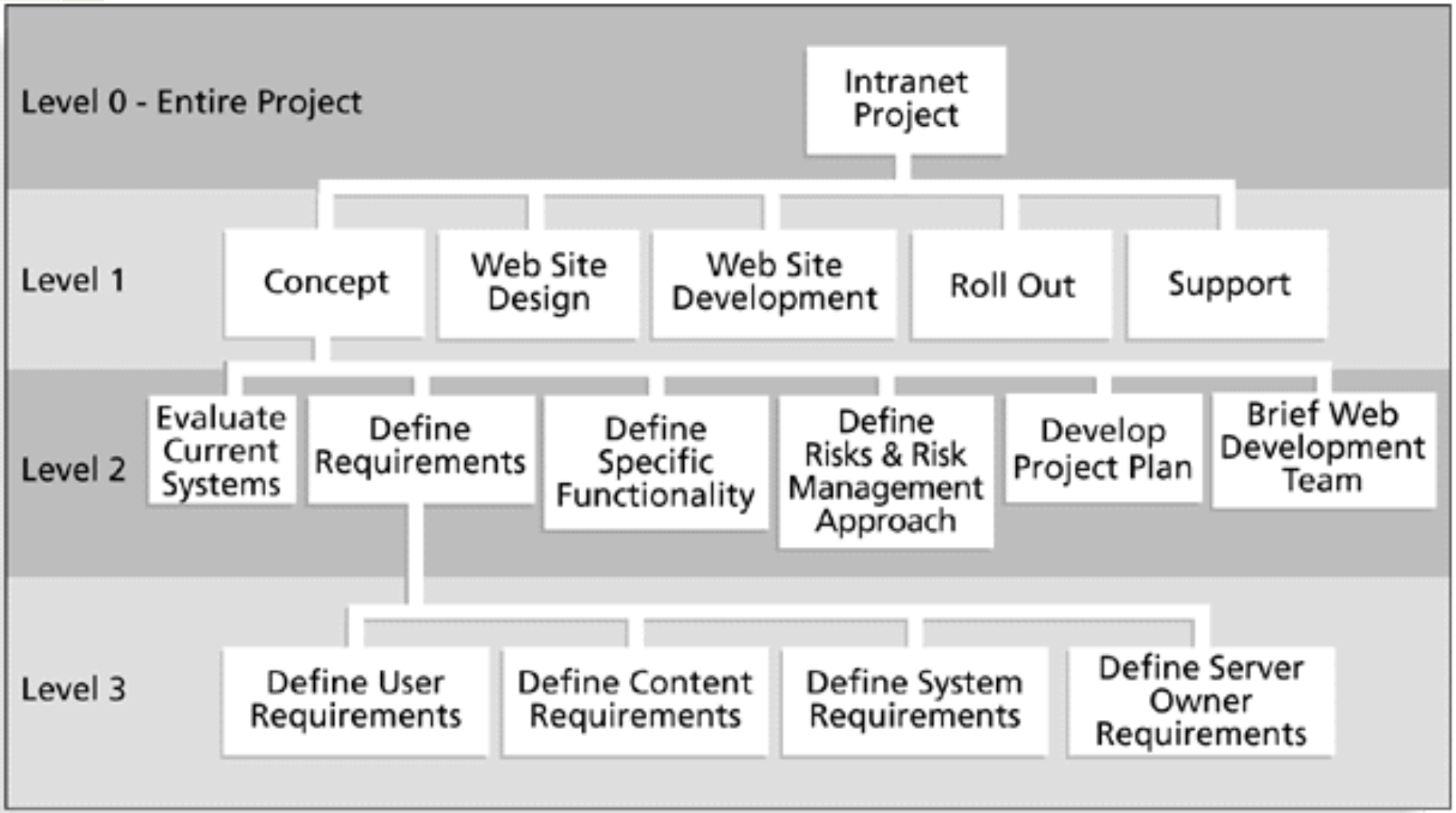
Types

- Process WBS
  - a.k.a Activity-oriented
  - Ex: Requirements, Analysis, Design, Testing
  - Typically used by PM
- Product WBS
  - a.k.a. Entity-oriented
  - Ex: Financial engine, Interface system, DB
  - Typically used by engineering manager
- Hybrid WBS: both above
  - This is not unusual
  - Ex: Lifecycle phases at high level with component or feature-specifics within phases
  - Rationale: processes produce products

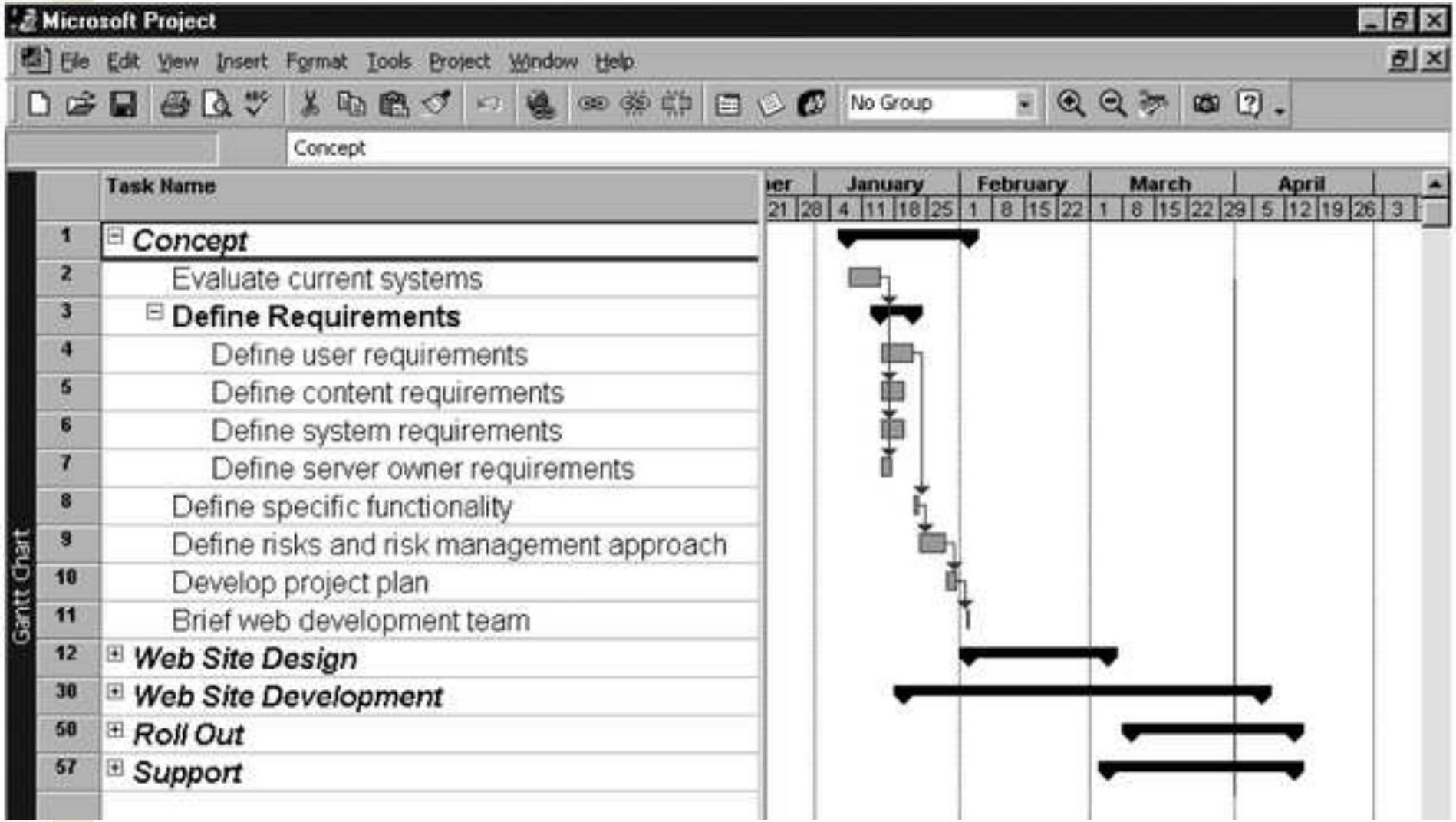
Product WBS


Intranet
Web Site Design
Site Map
Graphic Design
Programs
Home Page Design
Text
Images
Hyperlinks
Marketing Pages
Text
Images
Hyperlinks
Sales Pages
Text
Images
Hyperlinks

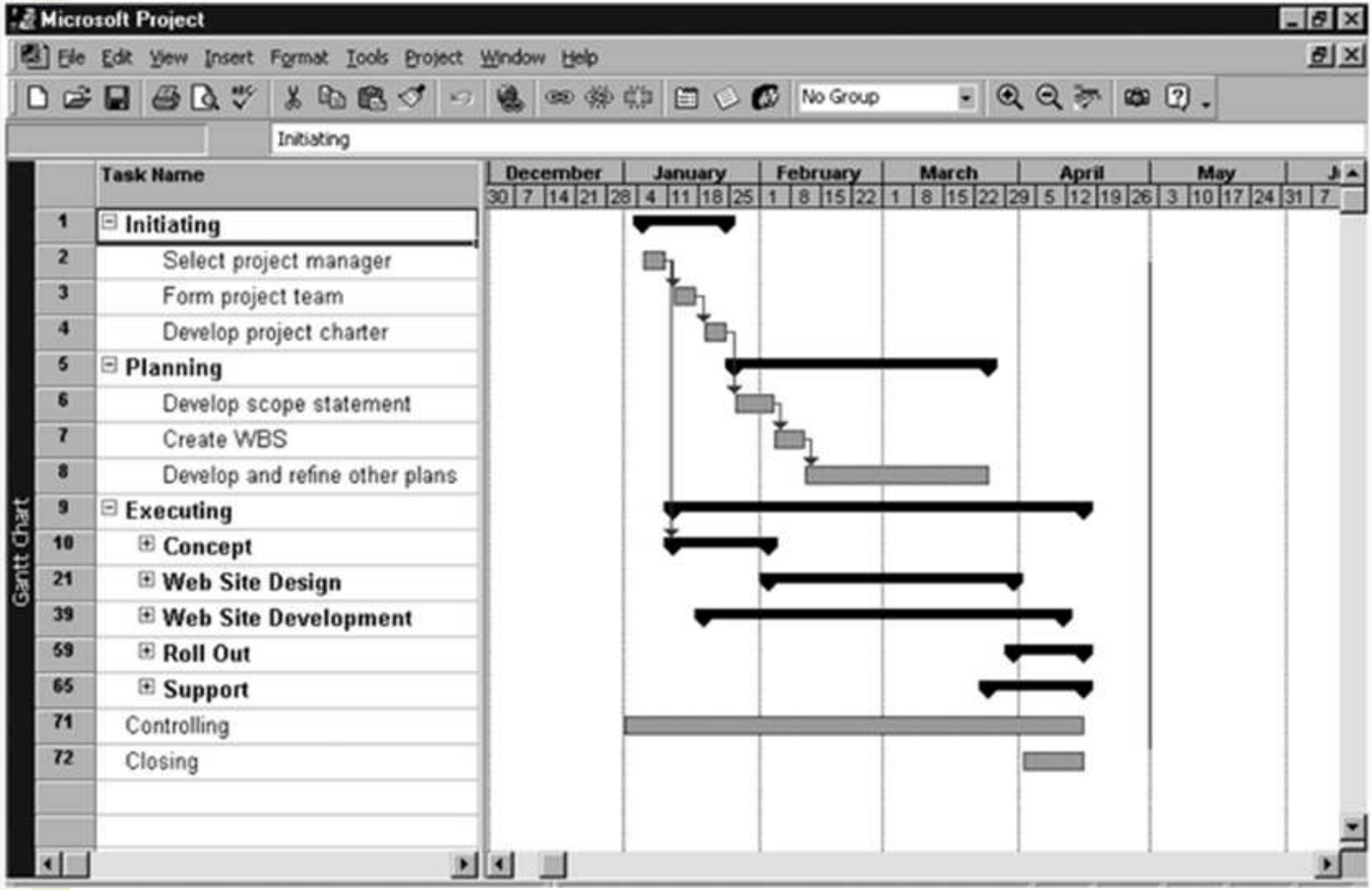
Process WBS


Level 0 - Entire Project
Intranet Project
Level 1
Concept
Web Site Design
Web Site Development
Roll Out
Support
Level 2
Evaluate Current Systems
Define Requirements
Define Specific Functionality
Define Risks & Risk Management Approach
Develop Project Plan
Brief Web Development Team
Level 3
Define User Requirements
Define Content Requirements
Define System Requirements
Define Server Owner Requirements

Outline WBS with Gantt Chart



WBS by PMI Process Groups



Less Frequently Used Alternatives

- Organizational WBS
  - Research, Product Design, Engineering, Operations
  - Can be useful for highly cross-functional projects
- Geographical WBS
  - Can be useful with distributed teams
  - NYC team, San Jose team, Off-shore team

Work Packages

- Generic term for discrete tasks with definable end results
- Typically the “leaves” on the tree
- The “one-to-two” rule
  - Often at: 1 or 2 persons for 1 or 2 weeks
- Basis for monitoring and reporting progress
  - Can be tied to budget items (charge numbers)
  - Resources (personnel) assigned
- Ideally shorter rather than longer
  - Longer makes in-progress estimates needed
  - These are more subjective than “done”
  - 2-3 weeks maximum for software projects
  - 1 day minimum (occasionally a half day)
  - Not so small as to micro-manage

Wrap Up (so far)

- List of Activities, not Things
- List of items can come from many sources
  - SOW, Proposal, brainstorming, stakeholders, team
- Describe activities using “bullet language”
  - Meaningful but terse labels
- All WBS paths do not have to go to the same level
- Do not plan more detail than you can manage

WBS & Chosen Lifecycle

- PM must map activities to chosen lifecycle
- Each lifecycle has different sets of activities
- Integral process activities occur for all
  - Planning, configuration, testing
- Operations and maintenance phases are not normally in plan (considered post-project)
- Some models are “straightened” for WBS
  - Spiral and other iterative models
  - Linear sequence several times
- Deliverables of tasks vary by chosen lifecycle

Techniques

- Top-Down
- Bottom-Up
- Analogy
- Brainstorming
  - Post-its on a wall
- Rolling Wave
  - 1st pass: go 1-3 levels deep
  - Gather more requirements or data
  - Add more detail later

Top-down

- Start at highest level
- Systematically develop increasing level of detail
- Best if
  - The problem is well understood
  - Technology and methodology are not new
  - This is similar to an earlier project or problem
- But is also applied in majority of situations

Bottom-up

- Start at lowest level tasks
- Aggregate into summaries and higher levels
- Cons
  - Time consuming
  - Needs more requirements complete
- Pros
  - Detailed

Analogy

- Base WBS upon that of a “similar” project
- Use a template
- Analogy also can be estimation basis
- Pros
  - Based on past actual experience
- Cons
  - Needs comparable project

Brainstorming

- Approach
  - Generate all activities you can think of that need to be done
  - Group them into categories
- Both Top-down and Brainstorming can be used on the same WBS
- Remember to get the people who will be doing the work involved (buy-in matters!)

WBS are Basis of Many Things

- Network scheduling
- Costing
- Risk analysis
- Organizational structure
- Control
- Measurement

Guidelines Part 1

- Should be easy to understand
- Some companies have corporate standards for these schemes
- Some top-level items, like Project Mgmt. are in WBS for each project
  - Others vary by project
- What often hurts most is what's missing
- Break down until you can generate accurate time & cost estimates
- Ensure each element corresponds to a deliverable

Guidelines Part 2

- How detailed should it be?
  - Not as detailed as the final MS-Project plan
  - Each level should have no more than 7 items
  - It can evolve over time
- What tool should you use?
  - Excel, Word, Project
  - Org chart diagramming tool (Visio, etc)
  - Specialized commercial apps
- Re-use a “template” if you have one

Introduction

- Very difficult to do, but needed often
- Created, used or refined during
  - Strategic planning
  - Feasibility study and/or SOW
  - Proposals
  - Vendor and sub-contractor evaluation
  - Project planning (iteratively)
- Basic process
  - Estimate the size of the product
  - Estimate the effort (man-months)
  - Estimate the schedule
  - NOTE: Not all of these steps are always explicitly performed

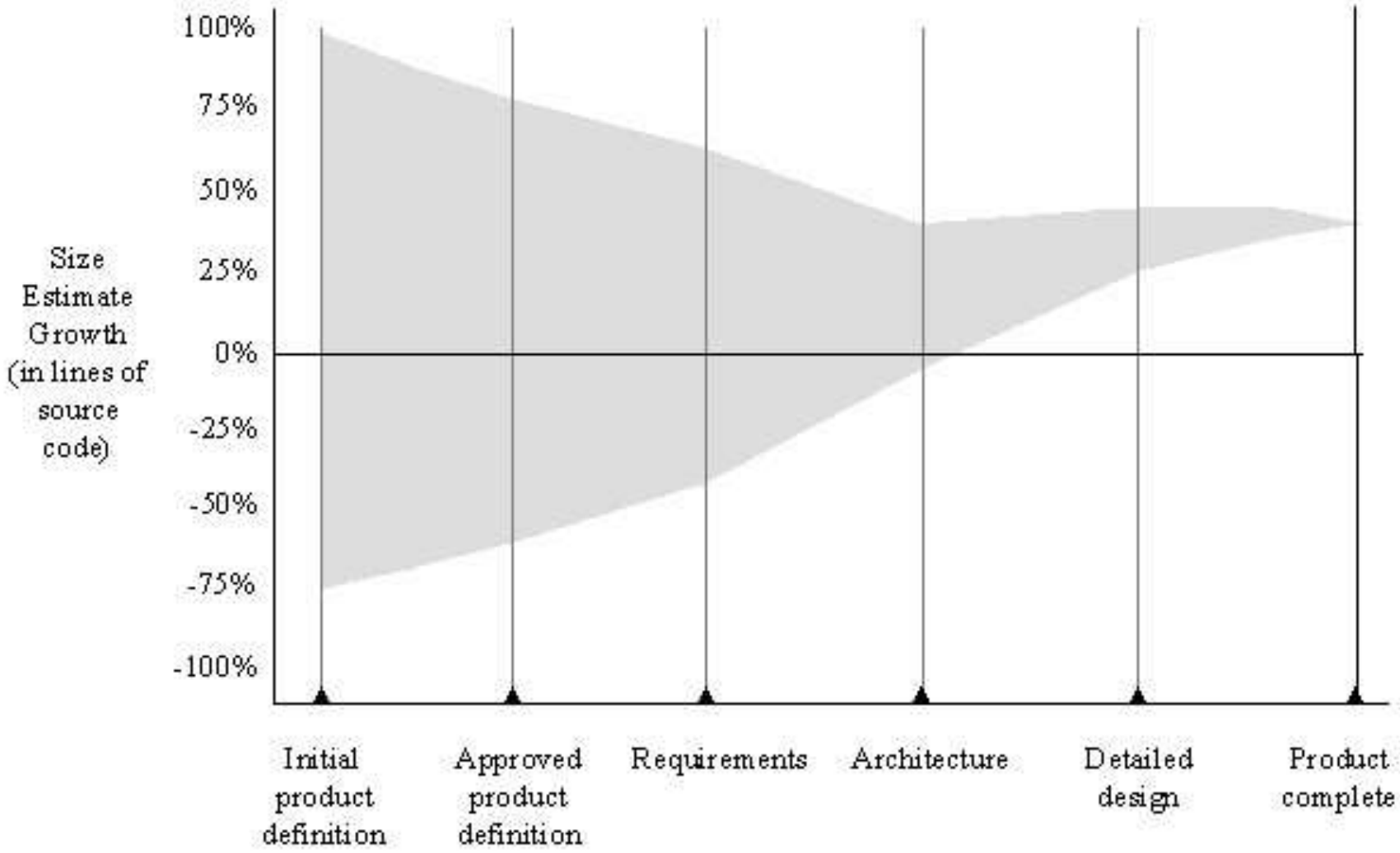
Introduction (cont.)

- Remember, an “exact estimate” is an oxymoron
- Estimate how long will it take you to get home from class this afternoon
  - On what basis did you do that?
  - Experience right?
  - Likely as an “average” probability
  - For most software projects there is no such ‘average’
- Most software estimations are off by 25-100%

Target vs. Committed Dates

- Target:
  - Proposed by business or marketing
  - Do not commit to this too soon!
- Committed:
  - Team agrees to this
  - After you've developed a schedule

Cone of Uncertainty



Size

- Small projects (10-99 FPs), variance of 7% from post-requirements estimates
- Medium (100-999 FPs), 22% variance
- Large (1000-9999 FPs) 38% variance
- Very large (> 10K FPs) 51% variance

Methodologies

- Top-down
- Bottom-up
- Analogy
- Expert Judgment
- Priced to Win
- Parametric or Algorithmic Method
  - Using formulas and equations

Top-down Estimation

- Based on overall characteristics of project
  - Some of the others can be “types” of top-down (Analogy, Expert Judgment, and Algorithmic methods)
- Advantages
  - Easy to calculate
  - Effective early on (like initial cost estimates)
- Disadvantages
  - Some models are questionable or may not fit
  - Less accurate because it doesn’t look at details

Bottom-up Estimation

- Create WBS
- Add from the bottom-up
- Advantages
  - Works well if activities well understood
- Disadvantages
  - Specific activities not always known
  - More time consuming

Expert Judgment

- Use somebody who has recent experience on a similar project
- You get a “guesstimate”
- Accuracy depends on their ‘real’ expertise
- Comparable application(s) must be accurately chosen
  - Systematic
- Can use a weighted-average of opinions

Estimation by Analogy

- Use past project
  - Must be sufficiently similar (technology, type, organization)
  - Find comparable attributes (ex: # of inputs/outputs)
  - Can create a function
- Advantages
  - Based on actual historical data
- Disadvantages
  - Difficulty ‘matching’ project types
  - Prior data may have been mis-measured
  - How to measure differences – no two exactly same

Priced to Win

- Just follow other estimates
- Save on doing full estimate
- Needs information on other estimates (or prices)
- Purchaser must closely watch trade-offs
- Price to lose?

Algorithmic Measures

- Lines of Code (LOC)
- Function points
- Feature points or object points
- Other possible
  - Number of bubbles on a DFD
  - Number of of ERD entities
  - Number of processes on a structure chart
- LOC and function points most common
  - (of the algorithmic approaches)
- Majority of projects use none of the above

Code-based Estimates

- LOC Advantages
  - Commonly understood metric
  - Permits specific comparison
  - Actuals easily measured
- LOC Disadvantages
  - Difficult to estimate early in cycle
  - Counts vary by language
  - Many costs not considered (ex: requirements)
  - Programmers may be rewarded based on this
    - Can use: # defects/# LOC
  - Code generators produce excess code

LOC Estimate Issues

- How do you know how many in advance?
- What about different languages?
- What about programmer style?
- Stat: avg. programmer productivity: 3,000 LOC/yr
- Most algorithmic approaches are more effective after requirements (or have to be after)

Function Points

- Software size measured by number & complexity of functions it performs
- More methodical than LOC counts
- House analogy
  - House's Square Feet ~= Software LOC
  - # Bedrooms & Baths ~= Function points
  - Former is size only, latter is size & function
- Three basic steps

Function Point Process

1. Count # of biz functions per category
   - Categories: outputs, inputs, db inquiries, files or data structures, and interfaces
2. Establish Complexity Factor for each and apply
   - Simple, Average, Complex
   - Set a weighting multiplier for each (0->15)
   - This results in the “unadjusted function-point total”
3. Compute an “influence multiplier” and apply
   - It ranges from 0.65 to 1.35; is based on 14 factors
4. Results in “function point total”
   - This can be used in comparative estimates

- For a tutorial: http://www.devdaily.com/FunctionPoints/

Wideband Delphi

- Group consensus approach
- Rand corp. (http://www.rand.org/) used original Delphi approach to predict future technologies
- Present experts with a problem and response form
- Conduct group discussion, collect anonymous opinions, then feedback
- Conduct another discussion & iterate until consensus
- Advantages
  - Easy, inexpensive, utilizes expertise of several people
  - Does not require historical data
- Disadvantages
  - Difficult to repeat
  - May fail to reach consensus, reach wrong one, or all may have same bias

Parametric Method Issues

- Remember: most projects you'll run into don't use these
- Which is 'normal', so don't be surprised
  - Or come-in to new job and say "Hey, let's use COCOMO"
- These are more effective on large projects
  - Where a past historical base exists
- Primary issue for most projects are
  - Lack of similar projects
    - Thus lack of comparable data

Code Reuse & Estimation

- Does not come for free
- Code types: New, Modified, Reused
- If code is more than 50% modified, it's "new"
- Reuse factors have wide range
  - Reused code takes 30% effort of new
  - Modified is 60% of new
- Integration effort with reused code almost as expensive as with new code

Effort Estimation

- Now that you know the “size”, determine the “effort” needed to build it
- Various models: empirical, mathematical, subjective
- Expressed in units of duration
  - Person-Months (or Man-Months, less politically correct)

Effort Estimation

- McConnell shows schedule tables for conversion of size to effort
- As with parametric size estimation, these techniques perform better with historical data
- Again, not seen in ‘average’ projects
- Often the size and effort estimation steps are combined (not that this is recommended, but is what often is done)
- “Commitment-Based” Scheduling is what is often done
  - Ask developer to ‘commit’ to an estimate (his or her own)

COCOMO

- COnstructive COst MOdel
- Allows for the type of application, size, and “Cost Drivers”
- Outputs in Person Months
- Cost drivers using High/Med/Low & include
  - Motivation
  - Ability of team
  - Application experience
- Biggest weakness?
  - Requires input of a product size estimate in LOC

Issues

- Quality estimations needed early but information is limited
- Precise estimation data available at end but not needed
  - Or is it? What about the next project?
- Best estimates are based on past experience
- Politics of estimation:
  - You may anticipate a “cut” by upper management
- For many software projects there is little or none
  - Technologies change
  - Historical data unavailable
  - Wide variance in project experiences/types
  - Subjective nature of software estimation

Over and Under Estimation

- Over estimation issues
  - The project will not be funded
    - Conservative estimates guaranteeing 100% success may mean funding probability of zero.
  - Parkinson's Law: Work expands to take the time allowed
  - Danger of feature and scope creep
  - Be aware of "double-padding": team member + manager
- Under estimation issues
  - Quality issues (short changing key phases like testing)
  - Inability to meet deadlines
  - Morale and other team motivation issues

Estimation Guidelines

- Estimate iteratively!
  - Process of gradual refinement
  - Make your best estimates at each planning stage
  - Refine estimates and adjust plans iteratively
  - Plans and decisions can be refined in response
  - Balance: too many revisions vs. too few

Know Your Deadlines

- Are they ‘Real Deadlines’?
  - Tied to an external event
  - Have to be met for project to be a success
  - Ex: end of financial year, contractual deadline, Y2K
- Or ‘Artificial Deadlines’?
  - Set by arbitrary authority
  - May have some flexibility (if pushed)

"Presentation"

- How you present the estimation can have huge impact
- Techniques
  - Plus-or-minus qualifiers
    - 6 months +/-1 month
  - Ranges
    - 6-8 months
  - Risk Quantification
    - +/- with added information
    - +1 month of new tools not working as expected
    - -2 weeks for less delay in hiring new developers
  - Cases
    - Best / Planned / Current / Worst cases
  - Coarse Dates
    - Q3 2009
  - Confidence Factors
    - April 1 – 10% probability, July 1 – 50%, etc.

Other Estimation Factors

- Account for resource experience or skill
  - Up to a point
  - Often needed more on the “low” end, such as for a new or junior person
- Allow for “non-project” time & common tasks
  - Meetings, phone calls, web surfing, sick days
- There are commercial ‘estimation tools’ available
  - They typically require configuration based on past data

Other Estimation Notes

- Remember: “manage expectations”
- Parkinson’s Law
  - “Work expands to fill the time available”
- The Student Syndrome
  - Procrastination until the last minute (cram)

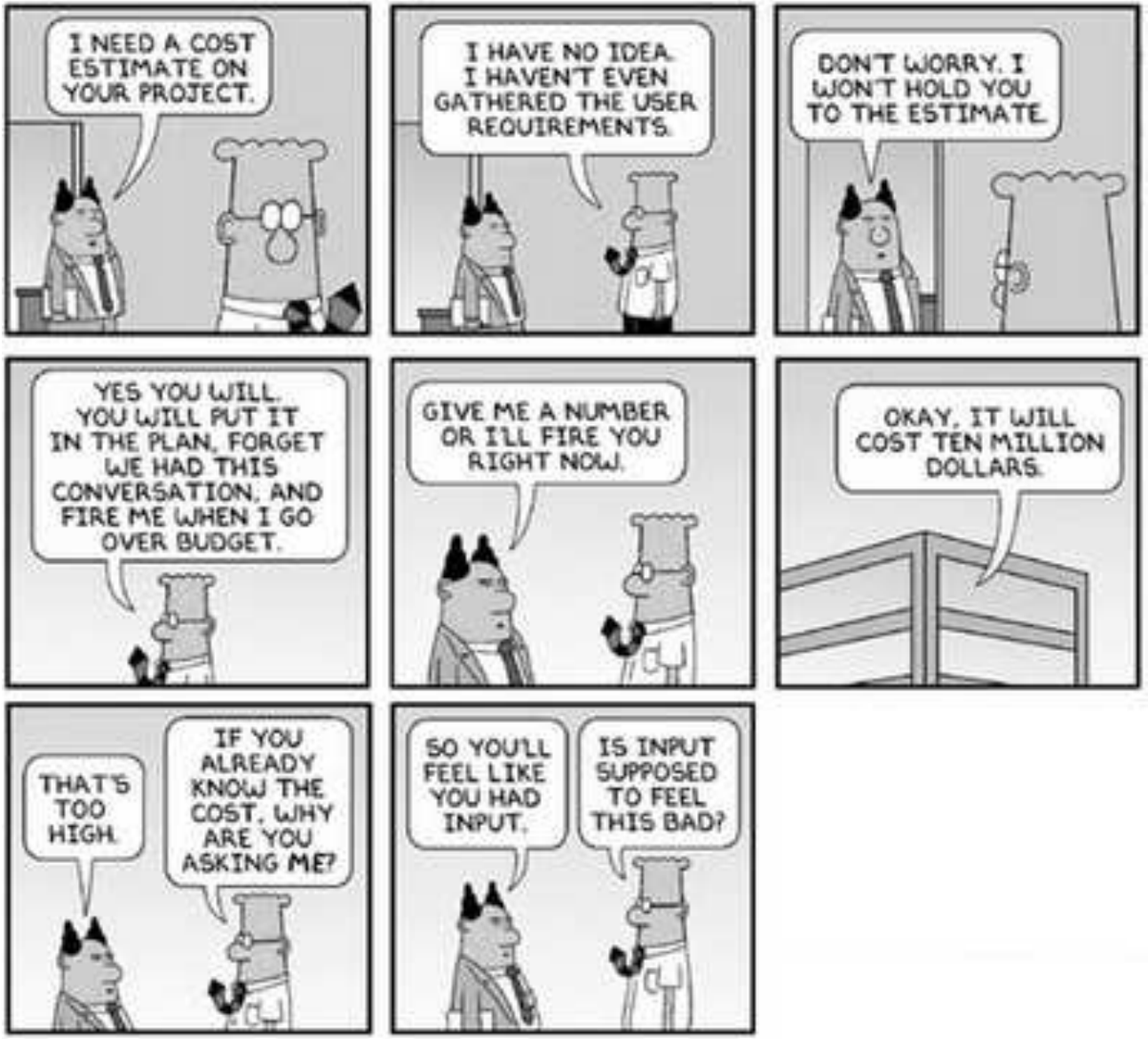
Optional Reading

- McConnell: 9, “Scheduling”
- Schwalbe: 5, “Project Time Management”
- URLs: See projectreference.com on “PERT/CPM”
  - http://www.projectreference.com/#PERT

Homework – 2: WBS

- Create a WBS for your project
  - Please think this through. You're the PM now!
- Guidelines
  - Do it at managerial level (see slide 14)
    - 4-7 nodes at 1st level
    - 2-5 nodes at 2nd level (per each node at 1st level)
    - You can go deeper at your discretion.
  - As we covered in class, you can use either a process, product or hybrid approach.
    - For most of your projects I suspect the process approach would work best at managerial level.
  - Follow the standard hierarchical numbering scheme for WBS structures.
- Format and Submission
  - Use the tool you prefer between Notepad/Word/Excel
  - Add homework-2 to the zip file you created for homework-1 and resubmit the entire package through the easychair Web site

Comix


I NEED A COST ESTIMATE ON YOUR PROJECT.
I HAVE NO IDEA. I HAVEN'T EVEN GATHERED THE USER REQUIREMENTS.
DON'T WORRY. I WON'T HOLD YOU TO THE ESTIMATE.
YES YOU WILL. YOU WILL PUT IT IN THE PLAN, FORGET WE HAD THIS CONVERSATION, AND FIRE ME WHEN I GO OVER BUDGET.
GIVE ME A NUMBER OR I'LL FIRE YOU RIGHT NOW.
OKAY, IT WILL COST TEN MILLION DOLLARS.
THAT'S TOO HIGH.
IF YOU ALREADY KNOW THE COST, WHY ARE YOU ASKING ME?
SO YOU'LL FEEL LIKE YOU HAD INPUT.
IS INPUT SUPPOSED TO FEEL THIS BAD?

Questions?