UNIVERSITY OF CAPE TOWN
Department of Computer Science

Advanced Software Design: Project Management

JonJon Clark
clrjon005@myuct.ac.za

slides and materials taken from prof edwin blake

16/7/15

2

OVERVIEW

Overview

Project Planning

People

Estimation and Metrics

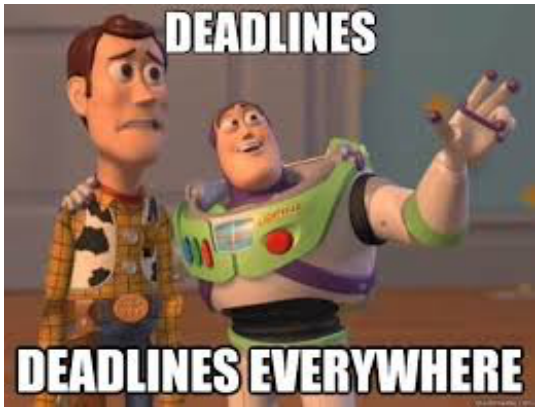
Scheduling

Risk

Conclusion

Project Management

- Concerned with activities involved in ensuring software is delivered on time and on schedule and in accordance with the requirements of the organisations developing and procuring the software.
- Project management is needed because software development is always subject to the **budget and schedule constraints** that are set by the organisation developing the software.



Project Management

- Firstly read Bennett Chapter 22 (online) Managing Object-Oriented Projects
- Systems development is a complex activity that requires careful project management.
- Inter-dependencies between the artefacts of software development
  - production has to be planned, monitored and co-ordinated
  - so that software development is efficient, effective and on time.
  - E.g. When would you start building the Chess AI?
- A software development project may involve many developers, some with specialized skills, who will be required at different times.
- Activities must follow a particular sequence.
  - For example, testing a system can only begin when at least some elements have been constructed
  - test scripts and test harnesses may be prepared earlier in the project.

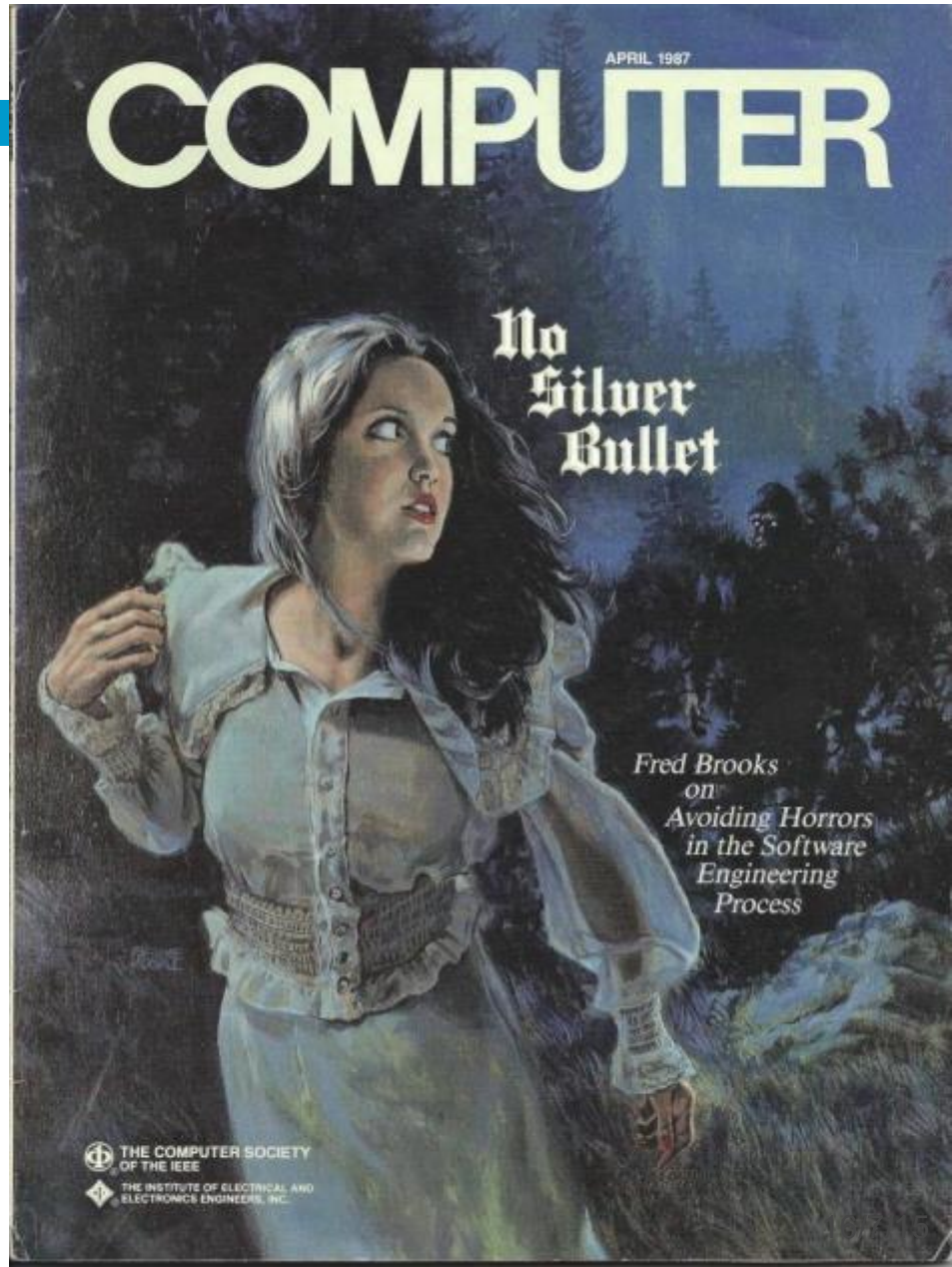
Warning: No silver bullet …

That is, *there ain't no technique that fixes the inherent complexity of software design and development*

- "No Silver Bullet" refers to Fred Brooks at IFIP Conf of 1986.
- Brooks, F.P., "No Silver Bullet: Essence and Accidents of Software Engineering" Computer, 20, 4. April 1987. 10–19

doi.ieeecomputersociety.org/10.1109/MC.1987.1663532

ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1663532



Fred Brooks: 2007

- Difficulties with building software can be divided (Aristotle) into
  - ***Essence***: conceptual structure of the software itself, quite apart from any realization;
  - ***Accidents****:* difficulties from aspects of the process of realizing the conceptual structure in executable form.

+ a lot of projects start out seemingly innocent and straightforward, and then in the dark of the moon they turn into monsters (werewolves)!

- Only if at least 9/10 of troubles are accidents can shrinking them zero give you an order of magnitude improvement.
  - And they are not!
- Therefore, any silver bullet (anything offering *10 improvement in productivity) must address the conceptual issues
  - may mean dealing with the concepts at a different level

Project Management Tools

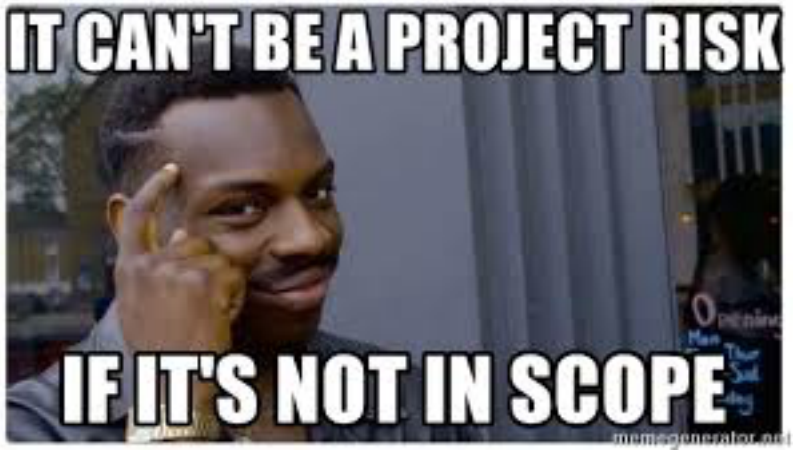
- Basecamp
- Freedcamp
- TeamGantt
- Asana
- Choose one of these for Capstone project!

Also – Slack, Overleaf etc…

PROJECT PLANNING

Scope of a Project

- the functions and features that are to be delivered to end-users
- the data that are input and output
- the content that is presented to users as a consequence of using the software
- the performance, constraints, interfaces and reliability that bound the system



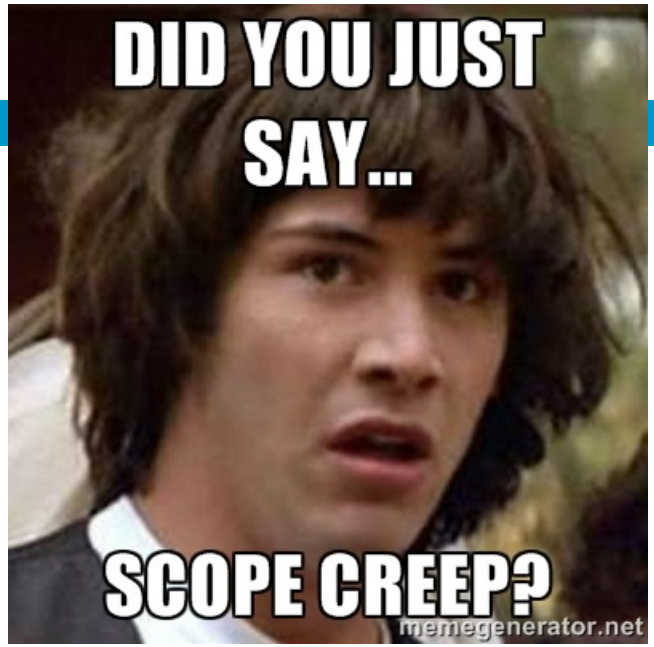
Defining Scope

- Scope is defined using one of two techniques
  - A narrative description of software scope is developed after communication with stakeholders.
  - A set of use-cases is developed. A use-case is scenario-based description of the user's interaction with the software from the user's point of view.
- Performance considerations encompass processing and response time requirements.
- Constraints identify limits placed on the software by external hardware, available memory, or other existing systems.

Determining Feasibility

- Given the scope, determine the resources required.
  - Cost
- Make sure you have an answer to the question:
  - Do we have the resources we need to build this software?
- Resources are:
  - people
  - hardware and software tools
  - reusable components

Project Planning
Goals — aka Scope



- Goals are derived from the needs and expectations of stakeholders:
  - project sponsor;
  - customer for the deliverables;
  - users of the project outputs;
  - project manager and project team.
- Find true needs that create real benefits
  - Prioritize them
- Create measurable goals

S – specific, significant, stretching

M – measurable, meaningful, motivational

A – agreed upon, attainable, achievable, acceptable, action-oriented

R – realistic, relevant, reasonable, rewarding, results-oriented

T – time-boxed, time-based, time-bound, timely, tangible, trackable

Project Planning Deliverables II

- Deliverables are a list of items that have to be delivered to meet the goals
  - Verifiable and specific
  - Can be report, equipment acquisition or Executable code module etc…
  - Has a stakeholder who needs it.
- Say when it is to be delivered
- Give quality standards, for example:
  - software runs!
  - document is according to specified format

Project Planning
Schedule (+more later)

- The effort required (in terms of time)
- The people required (and other resources)
- Update deliverables with this & work schedule out
- If the schedule is unrealistic you can justify some of the following:
  - renegotiate deadline — delay
  - additional resources — more expensive
  - reduce scope — fewer deliverables (Only thing that makes sense for capstone – carefully determine your scope!!!)

Project Planning
Supporting Plans

IV

Human Resource Plan

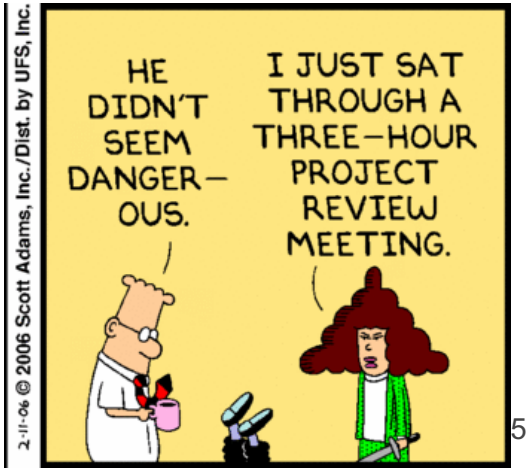
- Name key individuals & organisations: describe roles & responsibilities
- Describe the number and type of people needed
  - start dates, estimated duration and how to get them

Communications and Management Plan

- Who needs to be kept informed about the project?
- How they will receive the information?
  - weekly review meeting
  - progress reports
  - revised schedule

Risk Management Plan — see later

- Identify as many risks as possible
- be prepared if something bad happens.



The Range of Management Activities

People:

- Managers—Project Managers—Team Leaders—Software Team—Customers—End Users

Product

- Scope and decomposition

Process

- SDLC: Initiation →Analysis →Design →Construct →Test →Implement
- UP: Inception →Elaboration →Construction →Transition →Production

Project

- Size estimation; scheduling; risk management; tracking.

17

PEOPLE

Roles and Responsibilities

- It seems obvious that a team needs people with *different* skills
- Range of duties in a small project
  - Project management: Strategist, Leader, Politician, Project Facilitator, Administrator
  - Systems analysis: Stakeholder needs, Interaction Designer, Cost estimator
  - User interface designer, user stories
  - Architect: Application overview, performance
    - Middleware —software layer between the operating system and the applications on a distributed computer network
  - Specialists as required: database, games engine, mobile development, …
  - Documentation: Amanuensis

13 Essential Roles in small software development teams I

1. **Course Developer**: Preparation and coordination of training.
2. **Database Designer**: Essential to the process, mainly due to the specificity of its knowledge.
3. **Implementer**: Programs sub-systems and components that support the desired functionalities.
4. **Integrator** (lead programmer): Responsible for maintaining the implementers' awareness of the project context, for identifying the tasks to be undertaken and for appointing the person responsible for each one. Also responsible for the initial definition of the critical dates of the project and for developing a plan for the integration of the sub-systems, to allow the project manager to inform the client when each feature is expected to be available.
5. **Process Engineer**: Mainly concerned with the management of the development process, its adaptation to the organizational context and monitoring its implementation, in order to identify and implement possible process improvements..

13 Essential Roles in small software development teams II

6. **Project Manager**: Assume a global overview of the project through a detailed interaction with the internal and external participants. Must create the conditions for the project to achieve success, by ensuring timeliness and fulfilment of all commitments. Requires: basic knowledge in management; knowledge about the client's business domain; project management methodologies and negotiation skills.
7. **Project Reviewer**: This role cannot be considered critical, however, due to responsibilities related to the verification and approval of several artefacts produced by other participants, and possible conflict of interests, this person cannot have another role within the project.
8. **Software Architect**: Responsible for setting the technological foundation on which the project implementation should be based. The software architect is responsible for managing the technical risks.
9. **System Administrator**: Focused on ensuring the provision of the infrastructure needs (e.g., PCs for developers, servers, etc.)

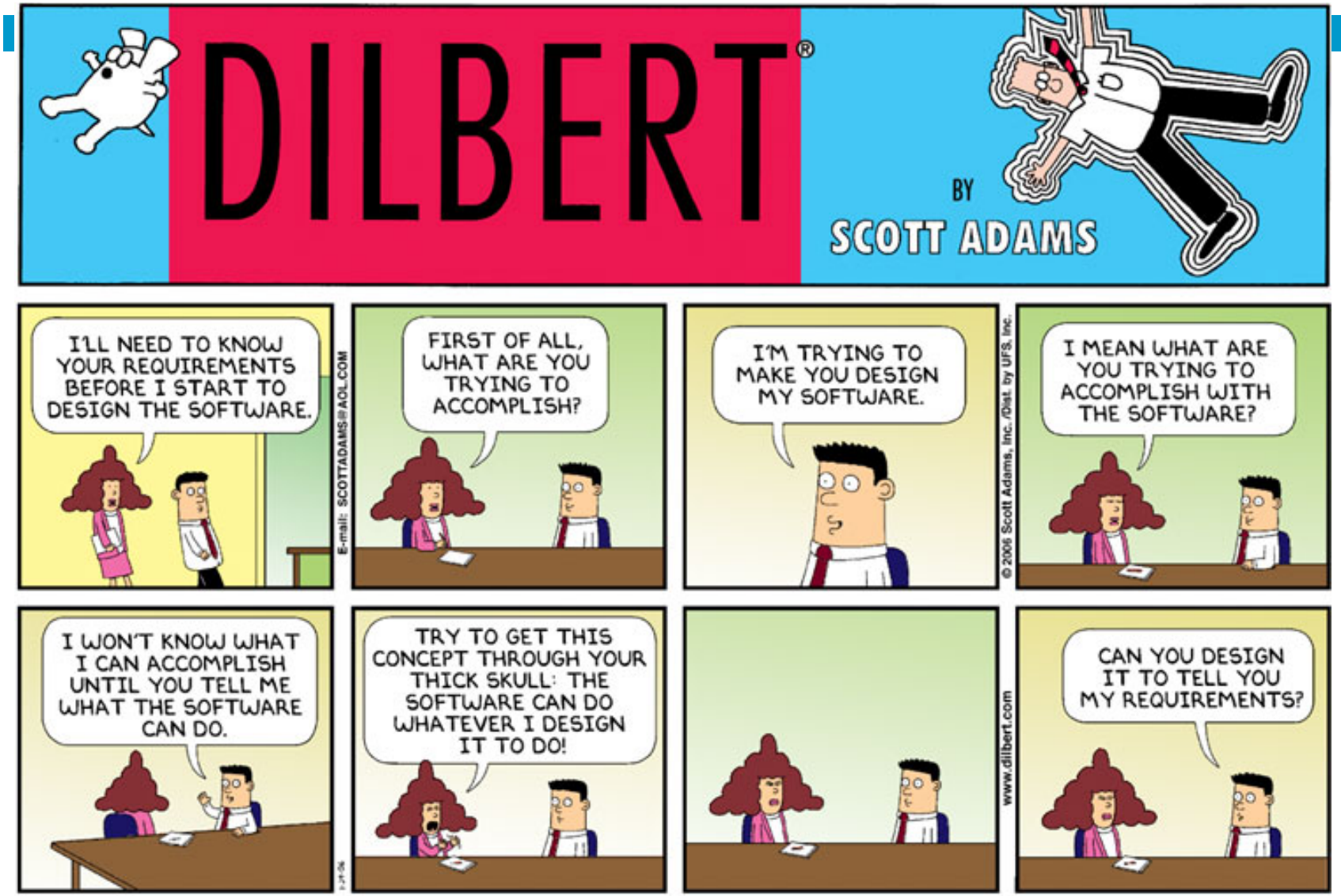
13 Essential Roles in small software development teams III

10. **System Analyst**: Scope management. Identify and document the requirements (functional or non-functional). Understand the client's business domain and to perceive the real motivations and relevance of the requirements.
11. **System Tester**: Entrusted with very different tasks, like review of documentation and testing behaviour.
12. **Test Manager**: Responsibility is to ensure the product quality by devising a plan for internal quality audits and implementation. Cannot have other roles, particularly with those roles related to the design and construction.
13. **User-Interface Designer:** The scope of this role in a project varies according to the nature of the artefacts to be developed.

Choosing People

- Information from candidates about their background and experience (résumé or CV)
  - Best evidence to judge suitability
- Information from interviewing candidates.
  - Mainly judge communication and social skills.
  - Subjective judgements: not reliable.
  - Can also ask to perform specific exercises.
- References and recommendations from people who have worked with them
  - Effective when you can rely on the people making the recommendation.

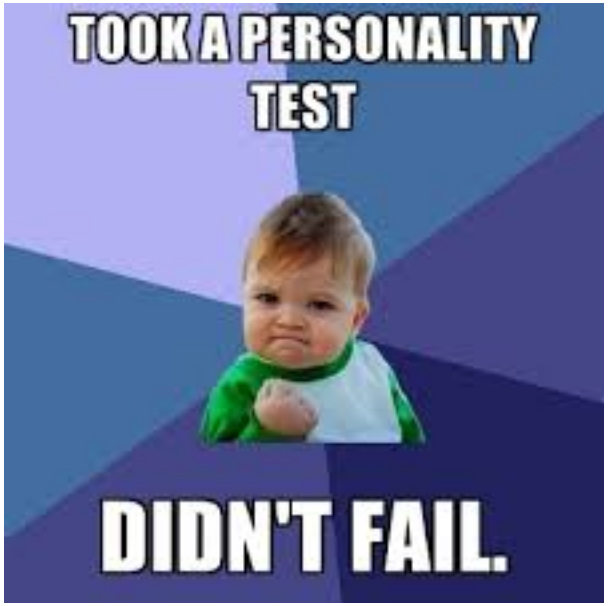
Dealing with People is the most important part of Project Management



Managing with different Personality Types

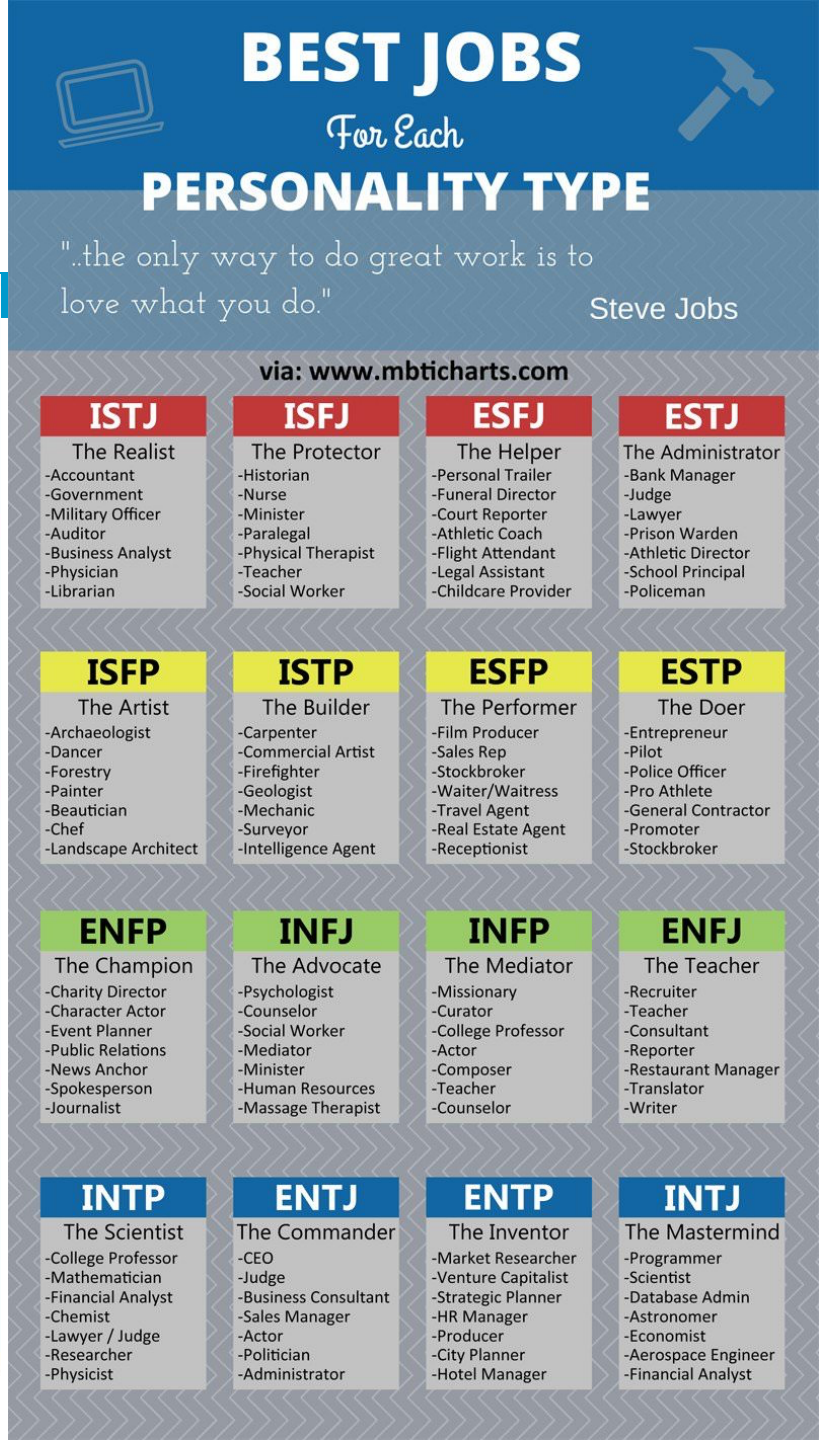
- You need to cater for:
  - backgrounds and personality styles of team members
  - management styles of customers and developers
- Realize that other people are not necessarily like you ...
  - and that not being a geek is OK too
- One standard is the Jung (1920s) + Myers-Briggs (1960s) personality model
  - see www.humanmetrics.com/cgi-win/JTypes2.asp
- Different kinds of personalities need different kinds of motivation, recognition and rewards.

- Take the test free online to see what you are! Take results with a pinch of salt!
- https://www.16personalities.com/free-personality-test



What are you?

ISTJ  Warren Buffet	ISFJ  Mother Theresa	INFJ  Mahatma Gandhi	INTJ  Mark Zuckerberg
ISTP  Steve Jobs	ISFP  Michael Jackson	INFP  J K Rowling	INTP  Jimmy Wales
ESTP  Donald Trump	ESFP  Larry Ellison	ENFP  Walt Disney	ENTP  Barack Obama
ESTJ  Steve Ballmer	ESFJ  Sam Walton	ENFJ  Oprah Winfrey	ENTJ  Bill Gates



Jung: two pairs of cognitive functions

1. The “irrational” (perceiving) functions: *sensing* and *intuition*
   - **How do you prefer to process information?**

   S — Sensing: prefer to deal with facts, what you know, to have clarity, or to describe what you see

   N — Intuition: prefer to deal with ideas, look into the unknown, to generate new possibilities or to anticipate what isn't obvious.
2. The “rational” (judging) functions: *thinking* and *feeling*
   - **How do you prefer to make decisions?**

   T — Thinking: prefer to decide on the basis of objective logic, using an analytic and detached approach

   F — Feeling: prefer to decide using values and/or personal beliefs, on the basis of what you believe is important or what you or others care about.

Jung: expression of Cognitive Functions

- **Where do you prefer to direct your energy?**

  E — Extravert: prefer to direct energy to deal with people, things, situations, or “the outer world”.

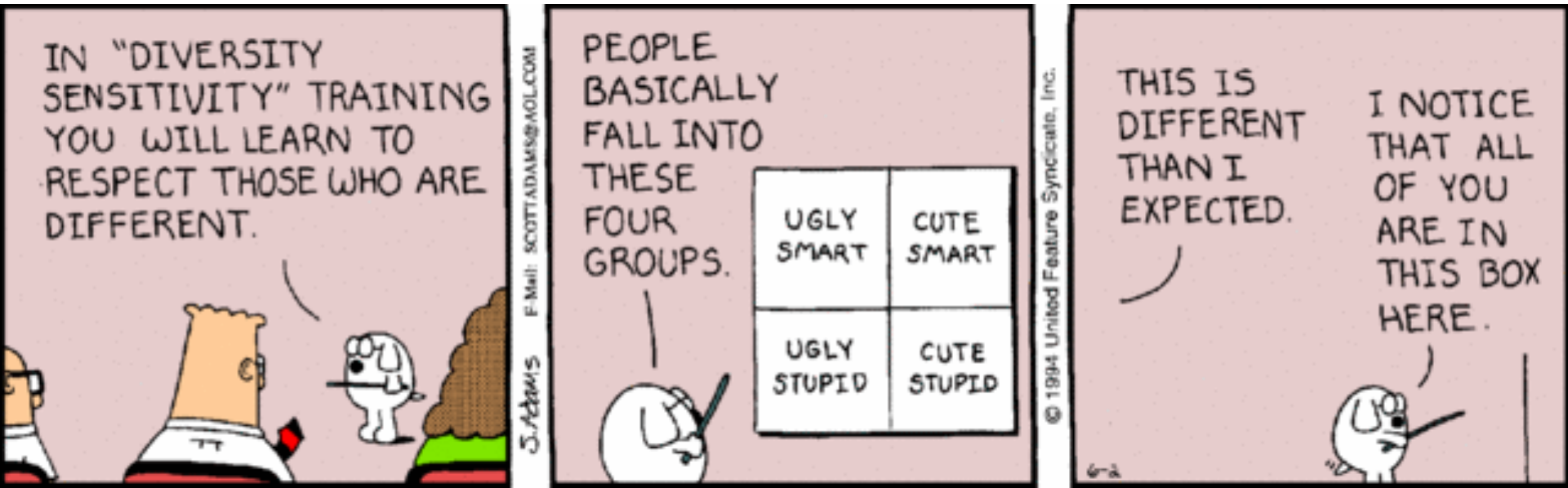
  I — Introvert: prefer to direct energy to deal with ideas, information, explanations or beliefs, or “the inner world”.

Meyers & Briggs: Lifestyle

- **How do you prefer to organise your life?**

J — Judging: prefer life to be planned, stable and organised (not to be confused with 'Judgmental'}

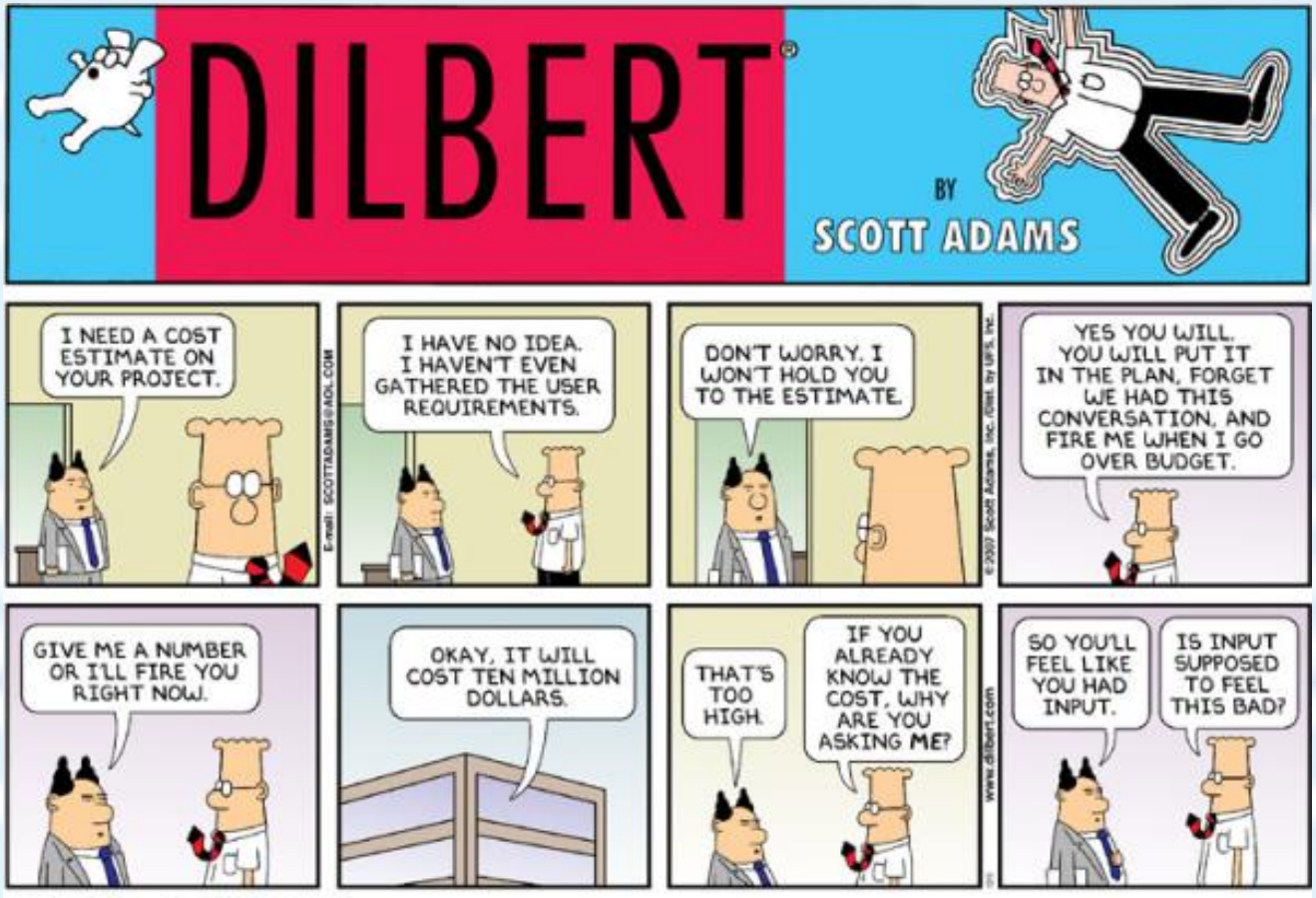
P — Perception: prefer to go with the flow, to maintain flexibility and respond to things as they arise.



Management and Team Success

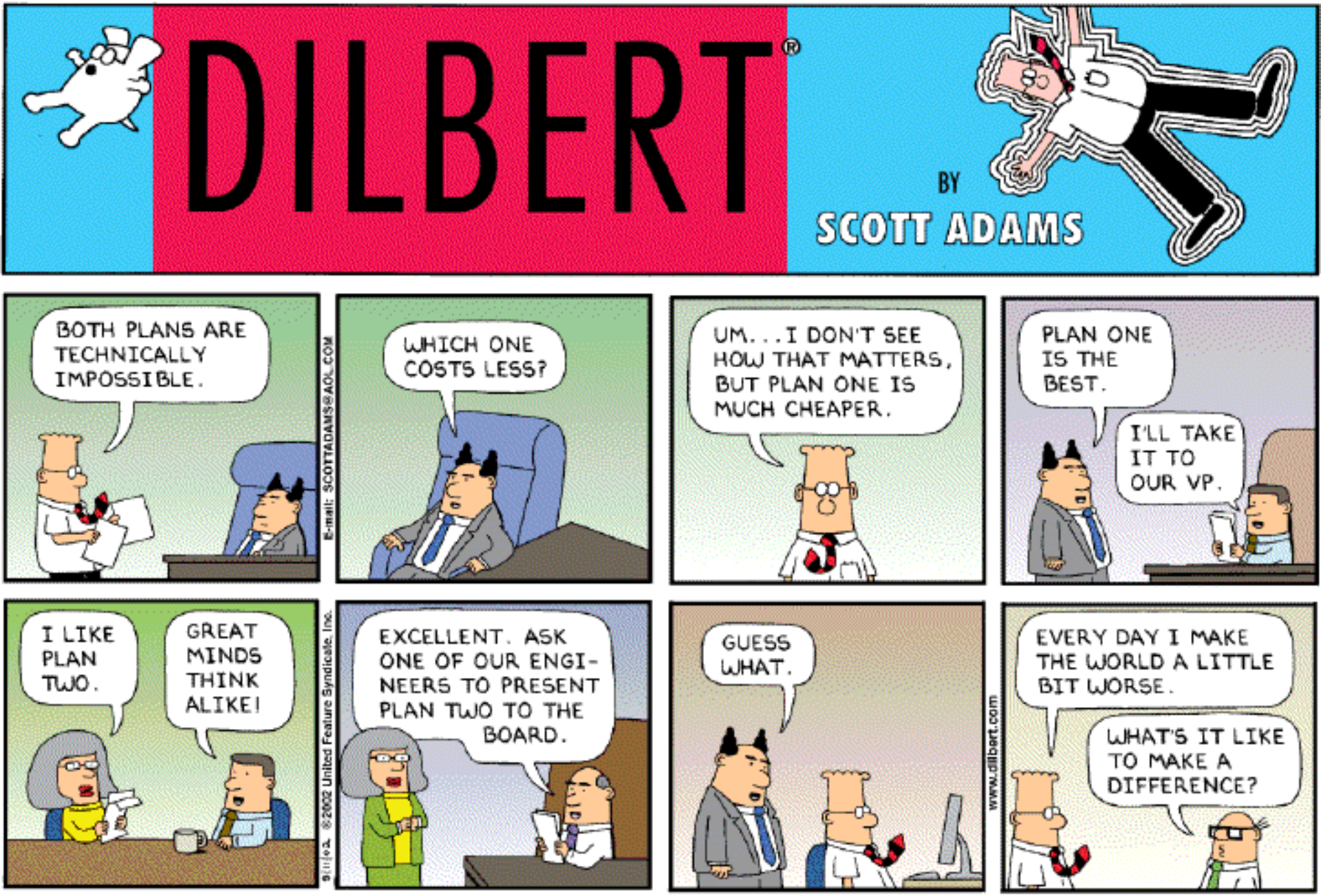
- Most software engineering is a group activity
  - Non-trivial software projects cannot be done by one person
  - People motivated by success of the group and their own personal goals
- Individual success depends on:
  - Ability and interest for work at hand
  - Experience and training with similar applications, development tools, programming languages
- Team success depends on:
  - **Ability to communicate** and express ideas in the team
    - be heard or be herded
  - Group interaction is a key determinant of group performance.
- Management skills
  - Limited flexibility in group composition: do the best with people available

ESTIMATION AND METRICS

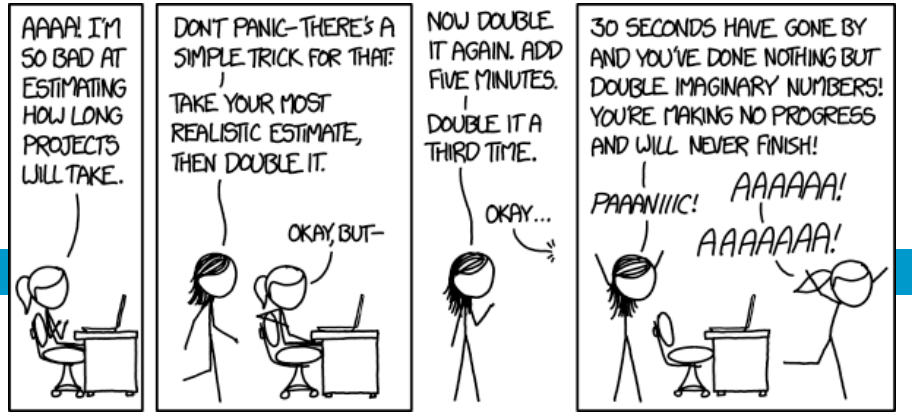


Milestones and Deliverables

- **Activity**: Task that takes time.
  - Duration: length of time needed for an activity.
  - Due date: date for completion of the activity
  - Precursor: activity which precedes others that depend on it
- **Milestone**:
  - Completion of an activity
  - Recognisable end-product of a task
    - hand over system for testing
  - Requires a formal, measurable output
    - "Coding 80% complete" is not adequate
- **Deliverable**:
  - A project result that is delivered (to customer)



Why Software Metrics?



- To plan and manage a software development project
  - Need to estimate the resources required for each of its constituent activities.
    - subjective perceptions of the activity
    - based upon measurements of size and complexity
      - activity itself
      - artefact that is produced.
  - Software metric measures some aspect of software development
    - project level—cost or duration
    - application level —size or complexity.

Software Metrics Characteristics I

- Process Metrics
  - process metrics measure some aspect of the development process
    - project cost to date
    - amount of time spent so far on the project

} these change all the time

- Product Metrics
  - product metrics measure some aspect of the software product.
    - analysis models
      - number of classes in an analysis class diagram.
    - test plans
    - program code

Software Metrics Characteristics II

- Result Metrics
  - measure outcomes
    - Current cost of a project (even though it will be modified tomorrow).
  - also known as *Control Metrics*
    - used to determine how management control should be exercised.
      - measurement of the current level of progress in the project is used to decide whether action is necessary to bring the project back onto schedule.

Software Metrics Characteristics III

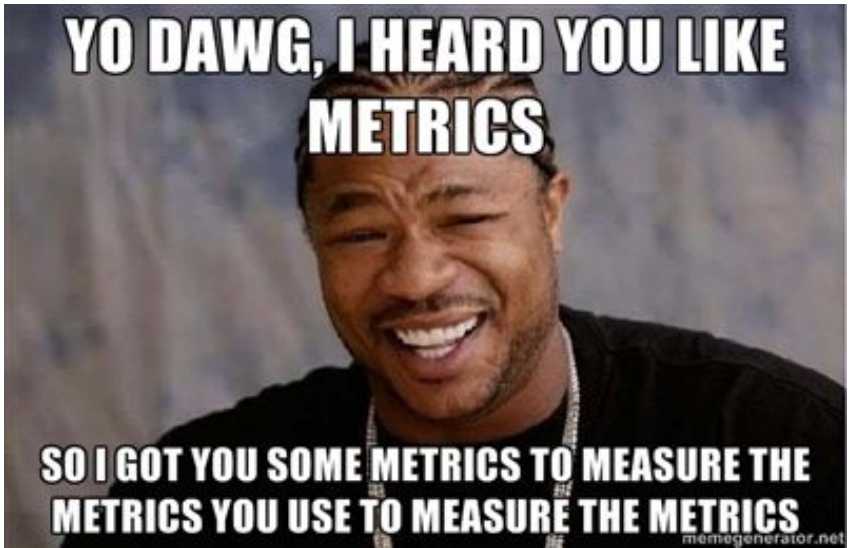
- Predictor Metrics
  - Quantify estimates for project resource requirements
    - Class size
      - a crude measure might be a simple count of attributes and operations
      - predictor because it can be used to predict the time that it will take to produce code
  - Also a measure of some aspect of a software product that is used to predict another aspect of the product or project progress.
    - Predict that the system will be difficult to maintain
    - Predict very low levels of reuse
      - change the design to improve the system.

Are Software Metrics Worth Anything? I

- Useful for prediction and resource estimation
  - Otherwise use of software metrics is rather limited.
- Validity of predictor metrics is based on 3 assumptions
  - You can measure something useful
  - That measure actually predicts something worthwhile
  - This relationship is real and can be expressed in a model or a formula.
- Size metrics can be used to estimate the resource requirement for a project provided that appropriate historical data is available to derive and validate the relationship.

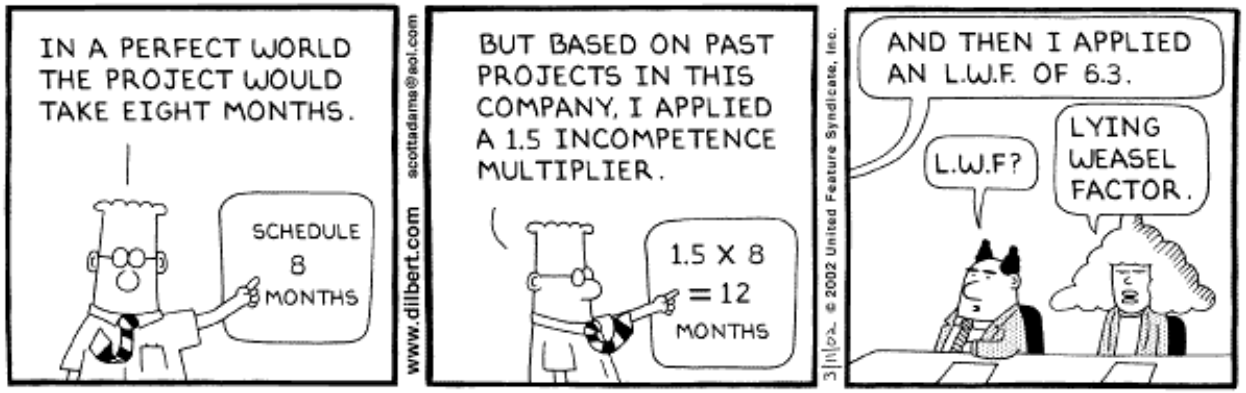
Are Software Metrics Worth Anything? II

- Generally software developers don't think metrics are important …
  - Focused on the delivery of the product on time.
  - Can be used to monitor the performance of the developers
    - This is a cause of concern to developers
    - Ethical issue and a management issue

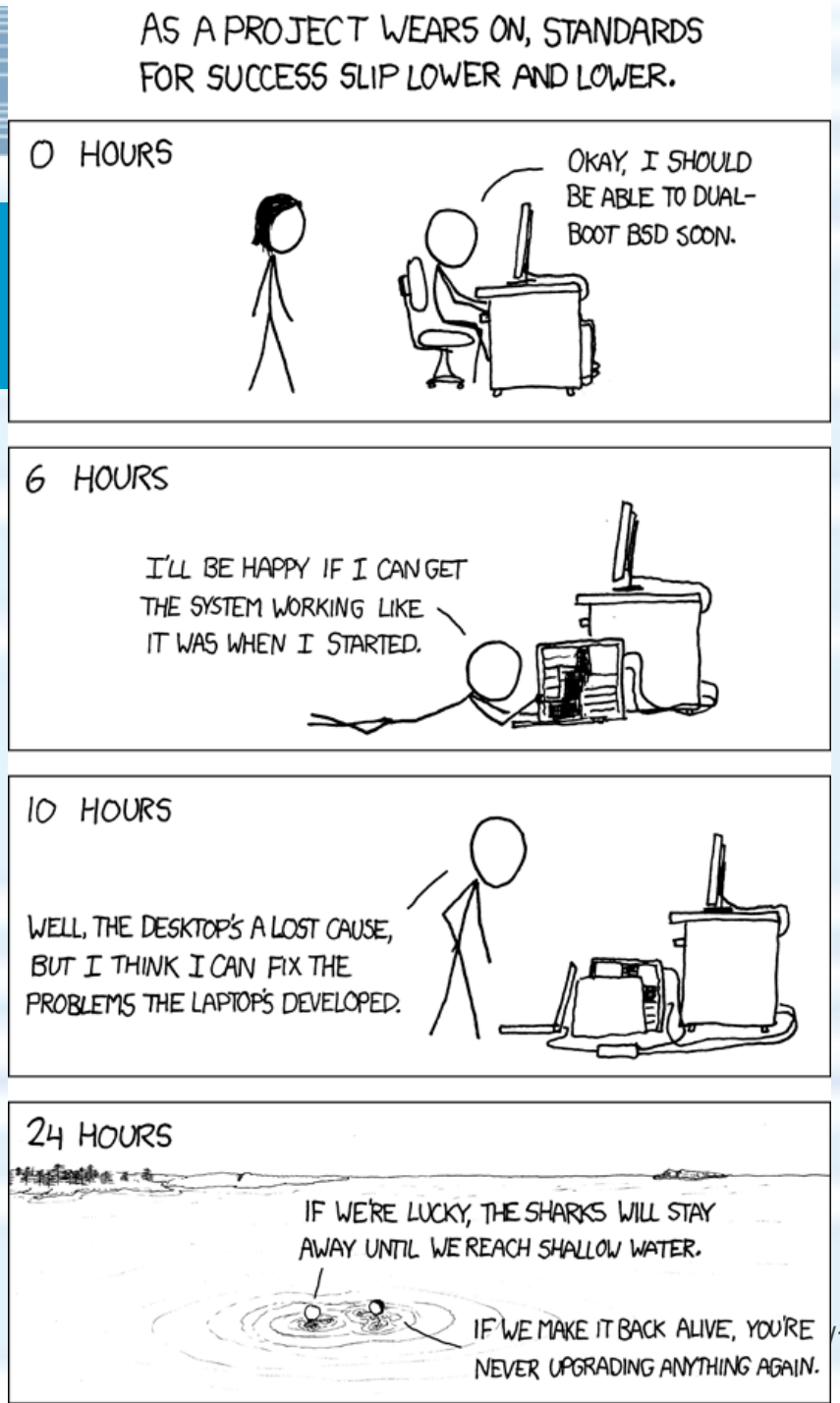


Metrics for Object-Oriented Development

- Ability of a package to absorb change is partly depends on ratio of abstract classes to all classes
  - 0 ⇒ package only has concrete classes and is difficult to change.
  - 1 ⇒ no concrete classes at all (easy to change!)
- Application size.
  - the number of use cases
  - the number of domain classes
  - multiplying factors that reflect the complexity of the user interface.
- Class size
  - number of attributes
  - number of operations
  - size of operations

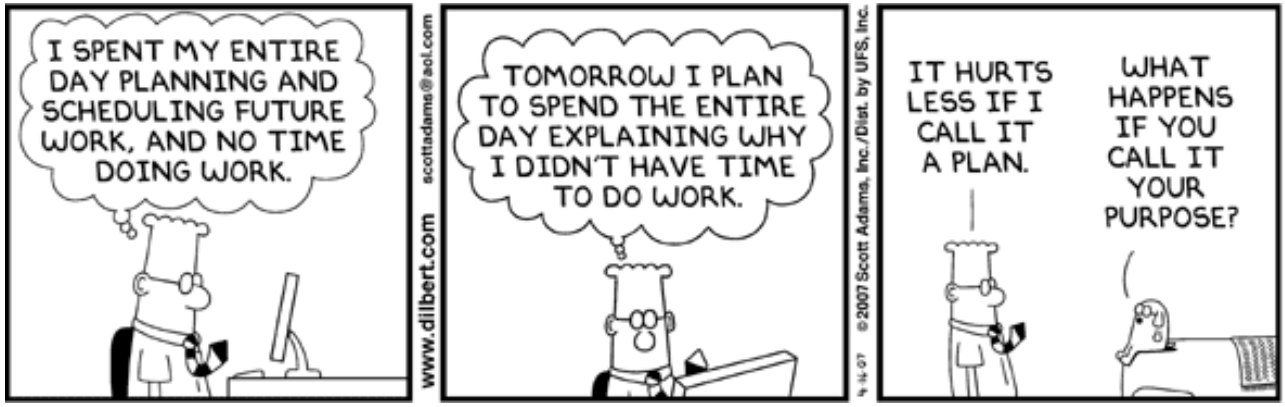


SCHEDULING



Project Scheduling

- Split the work in a project into separate tasks
  - Minimize task dependencies where one task waits for another to complete
- Estimate the calendar time needed to complete each task
  - Split up if much longer than 1 week (never more than 2 months)
  - Make tasks concurrent to make optimal use of workforce.
- Estimate the effort required
  - Who will work on the tasks
  - Resources needed to complete each task
- Mostly needs project manager's intuition and experience



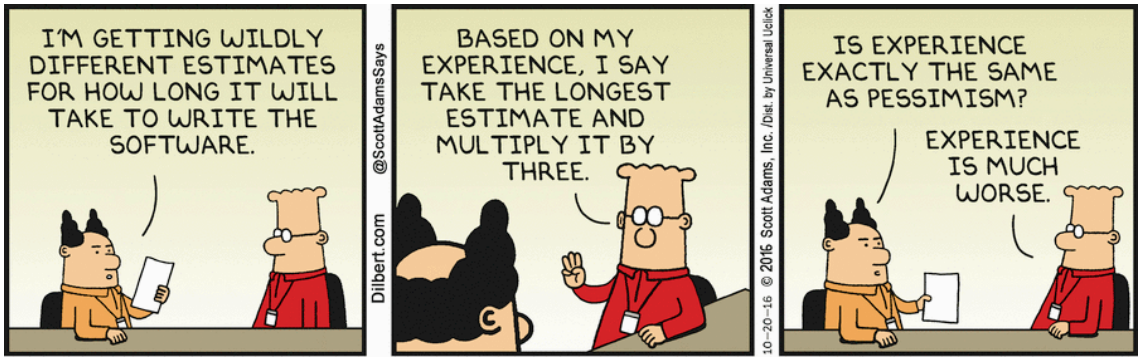
Project Scheduling Process

Software requirements and design information

Revise and update

Identify Activities → Identify Activity Dependencies → Estimate Resources for Activities → Allocate People to Activities → Create Project Charts

Bar charts describing the project schedule



non-agile, from Figure 23.4 in Sommerville

Critical Path or Network Analysis

- related to PERT charts
- Minimum overall duration of the project according to the estimates depends on the *critical path(s)*
- Any delay of a task on a critical path delays the whole project

What is Network Analysis?

- Project tasks (activities):
  - Are often interdependent
  - But need to be done in parallel for teamwork to be effective
- Task networks are graphical depictions of task dependence
- Network analysis is a project planning method that:
  - Determines the critical path
  - Establishes “most likely” time estimates
  - Calculates boundaries to stop project slippage

Terminology

Earliest Start/Finish

- Earliest a task can begin/end if all preceeding tasks are completed in the shortest time

Latest Start/Finish

- Latest a task can begin/end without delaying the minimum project completion time

Critical Path

- Chain that determines overall project duration
- Can be multiple critical paths (not just one)

Slack (Float)

- The amount of surplus time or leeway allowed while still maintaining the critical path

Tasks for a Multiuser 3D Meeting Place I

#	Activity	Description	Len	Depend
1	Access	provide access through an account, username, password and access privileges to system.		
1.1	User's status	keep a database of users status (online, available, busy, etc.).	30	
1.2	Persistence	to keep a database of all avatars, objects, their positions and states; thus providing "persistence" in the virtual world for users who come and leave.	180	1.1
1.3	Inventory	keep a database of users virtual inventory and support the addition, removal and exchange of items in the inventory.	10	1.2
2	Logging	log all events in world		
2.1	Log user events	log user status in world	10	1.3
2.2	Log avatar events	log position of avatars	10	3.4
2.3	Log object events	log position of objects.	10	4.3
2.4	Log communication	log all communication that occurs in VE.	100	2.1,2.2,2.3

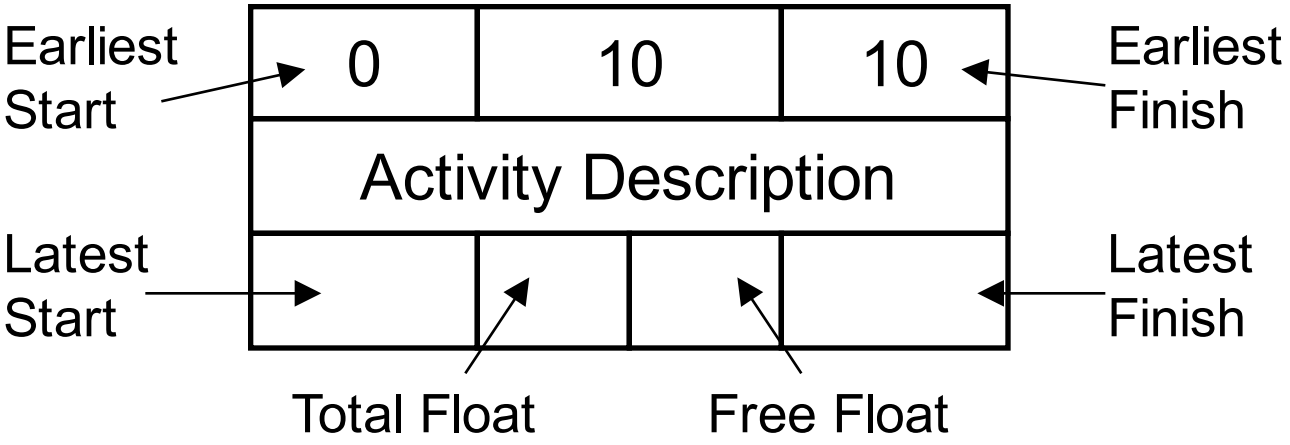
Tasks for a Multiuser 3D Meeting Place II

#	Activity	Description	Len	Depend
3	Avatar	provide user with a virtual representation of themselves in world.		
3.1	Avatar customization	provide user with the opportunity to easily customize avatar.	45	
3.2	Avatar Gesturing	provide user with the ability to express themselves using animated avatar gestures.	30	3.1
3.3	Avatar mood indicator	provide user with the ability to express their moods through visual representations (iconic, color, facial expressions ?).	160	3.2,1.1
3.4	Avatar proxemics	manage optimal proximity and gaze of groups of users engaging in communication.	20	3.3
4	Interactions	provide users with the ability to interact with VE, other users, their avatars and objects in VE.		
4.1	Manipulate objects	provide users with the ability to move, rotate, resize objects in VE in intuitive way.	20	
4.2	Operate Objects	provide users with the ability to operate interactive objects that have been scripted or have a menu interface.	120	4.1,1.1
4.3	Operate on Avatar	provide users with the ability to operate (issue commands) on other avatars through the menu interface.	60	1.2,4.2

Network Analysis

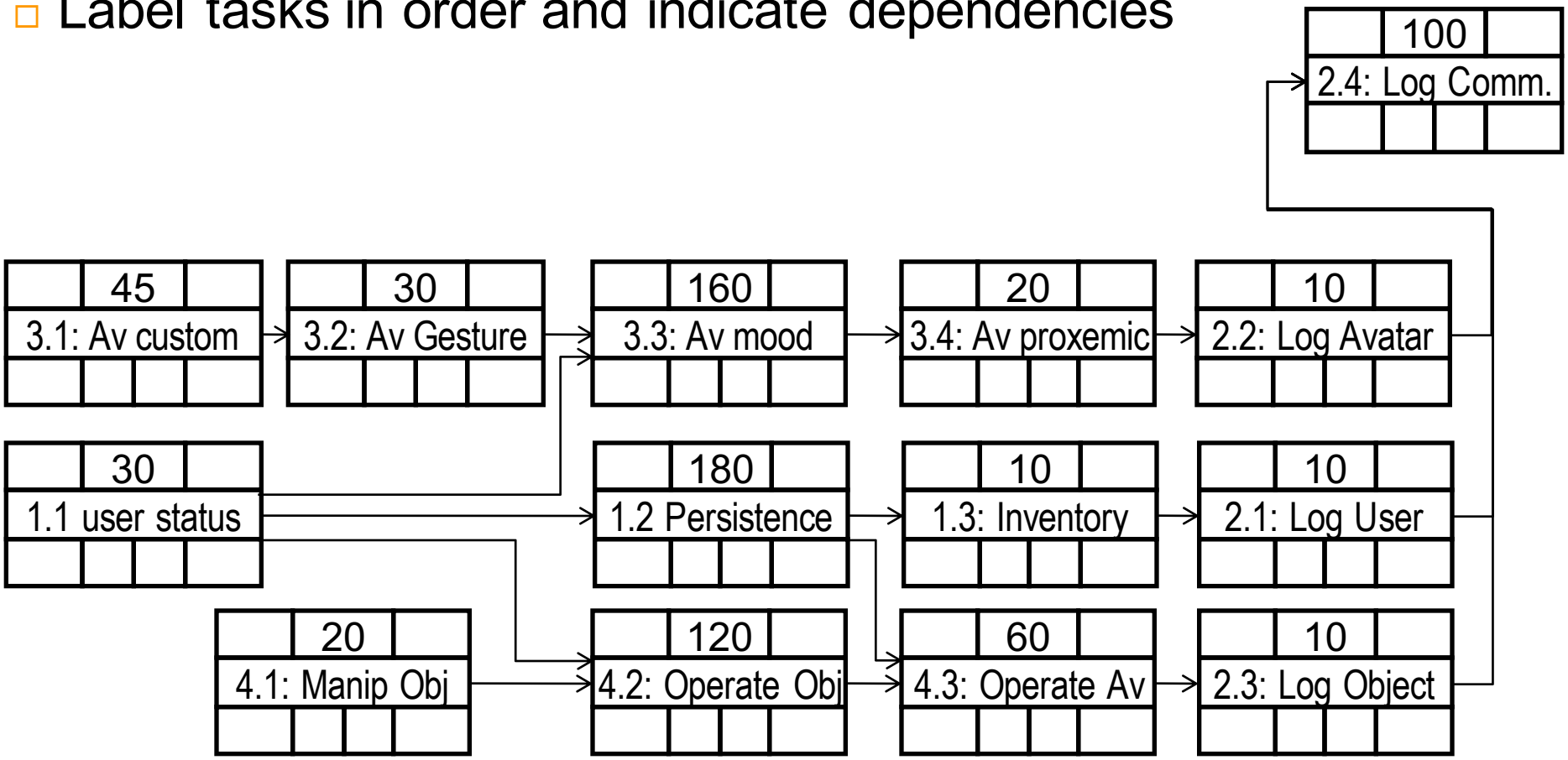
Tasks are shown as boxes

Sequence constraints are lines connecting the boxes

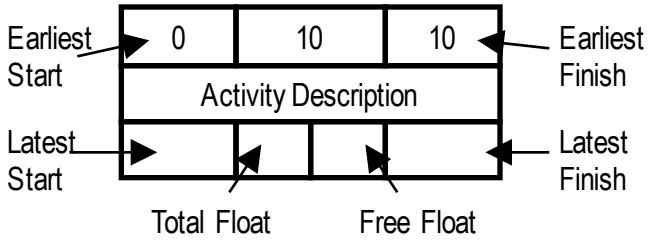


Analysis: Step #1

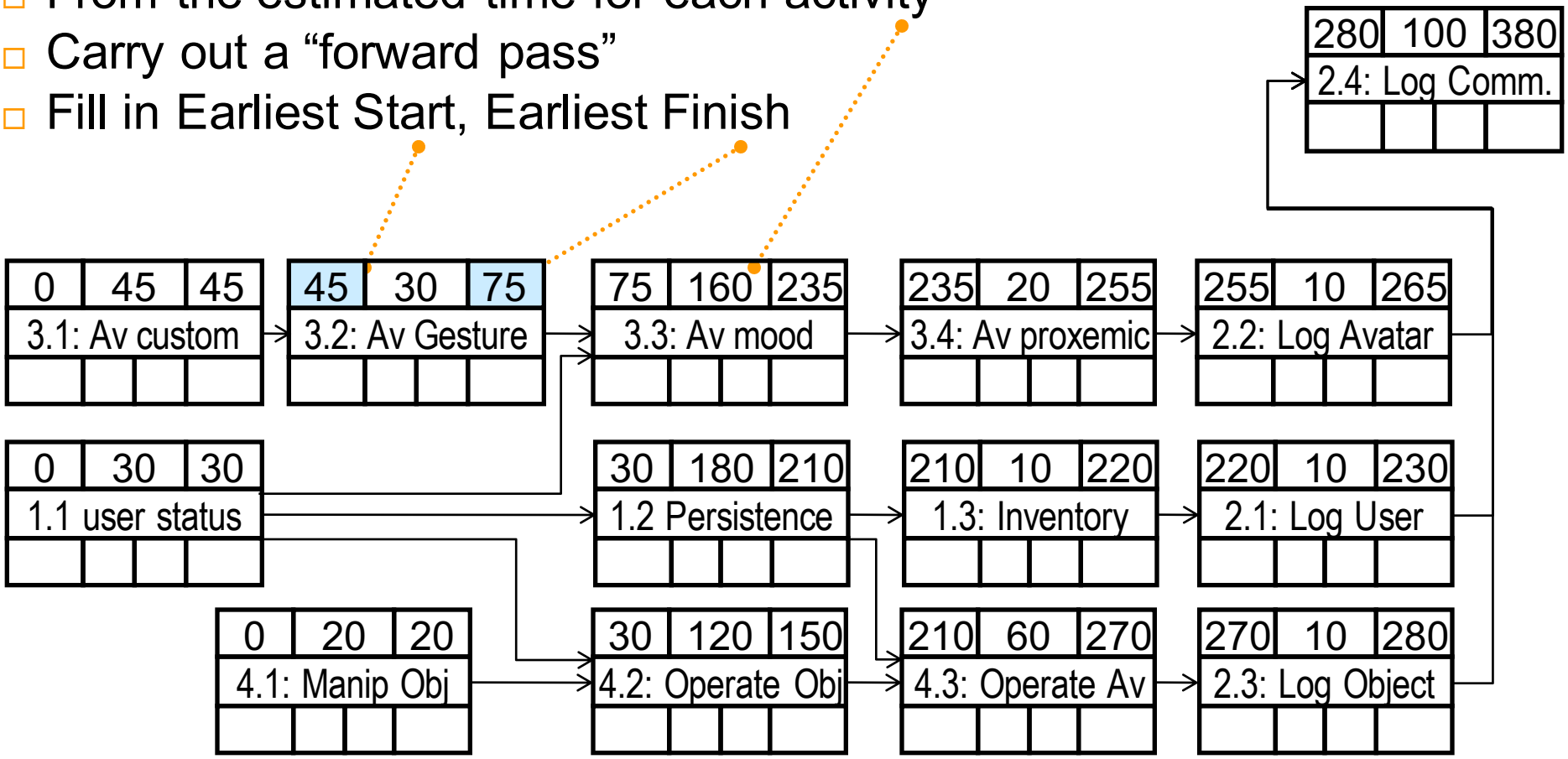
- Label tasks in order and indicate dependencies



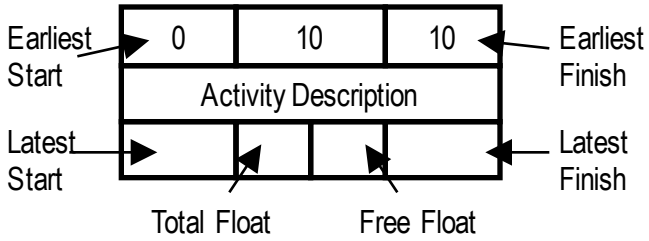
Analysis: Step #2



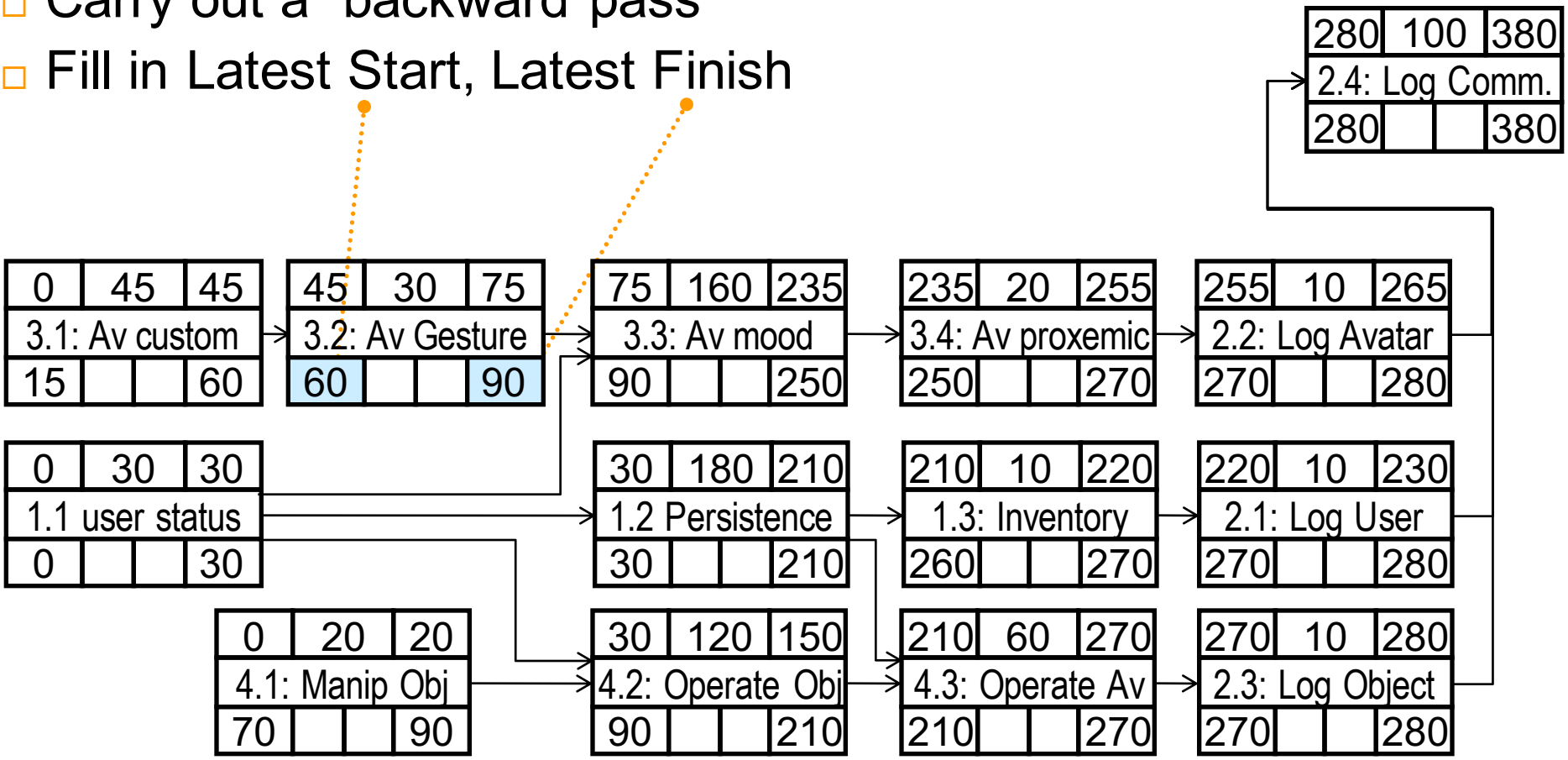
- From the estimated time for each activity
- Carry out a “forward pass”
- Fill in Earliest Start, Earliest Finish



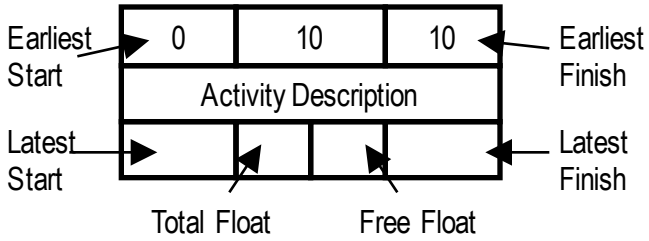
Analysis: Step #3



- Carry out a “backward pass”
- Fill in Latest Start, Latest Finish



Analysis: Step #4



- Fill in Total Float
  - Time by which an activity may be delayed without affecting the final completion date

280	100	380
2.4: Log Comm.		
280	0	380

0	45	45
3.1: Av custom		
15	15	60

45	30	75
3.2: Av Gesture		
60	15	90

75	160	235
3.3: Av mood		
90	15	250

235	20	255
3.4: Av proxemic		
250	15	270

255	10	265
2.2: Log Avatar		
270	15	280

0	30	30
1.1 user status		
0	0	30

30	180	210
1.2 Persistence		
30	0	210

210	10	220
1.3: Inventory		
260	50	270

220	10	230
2.1: Log User		
270	50	280

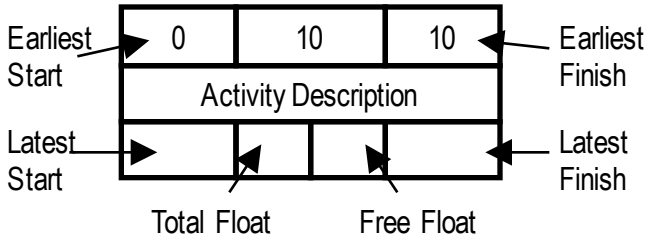
0	20	20
4.1: Manip Obj		
70	70	90

30	120	150
4.2: Operate Obj		
90	60	210

210	60	270
4.3: Operate Av		
210	0	270

270	10	280
2.3: Log Object		
270	0	280

Analysis: Step #5



- Fill in Free Float
  - Amount of time an activity can expand without affecting the start or finish of any other activity
- FF = ES (next) – EF (current)

0	45	45
3.1: Av custom		
15	15	0	60

45	30	75
3.2: Av Gesture		
60	15	0	90

75	160	235
3.3: Av mood		
90	15	0	250

235	20	255
3.4: Av proxemic		
250	15	0	270

255	10	265
2.2: Log Avatar		
270	15	15	280

280	100	380
2.4: Log Comm.		
280	0	0	380

0	30	30
1.1 user status		
0	0	0	30

30	180	210
1.2 Persistence		
30	0	0	210

210	10	220
1.3: Inventory		
260	50	0	270

220	10	230
2.1: Log User		
270	50	50	280

0	20	20
4.1: Manip Obj		
70	70	10	90

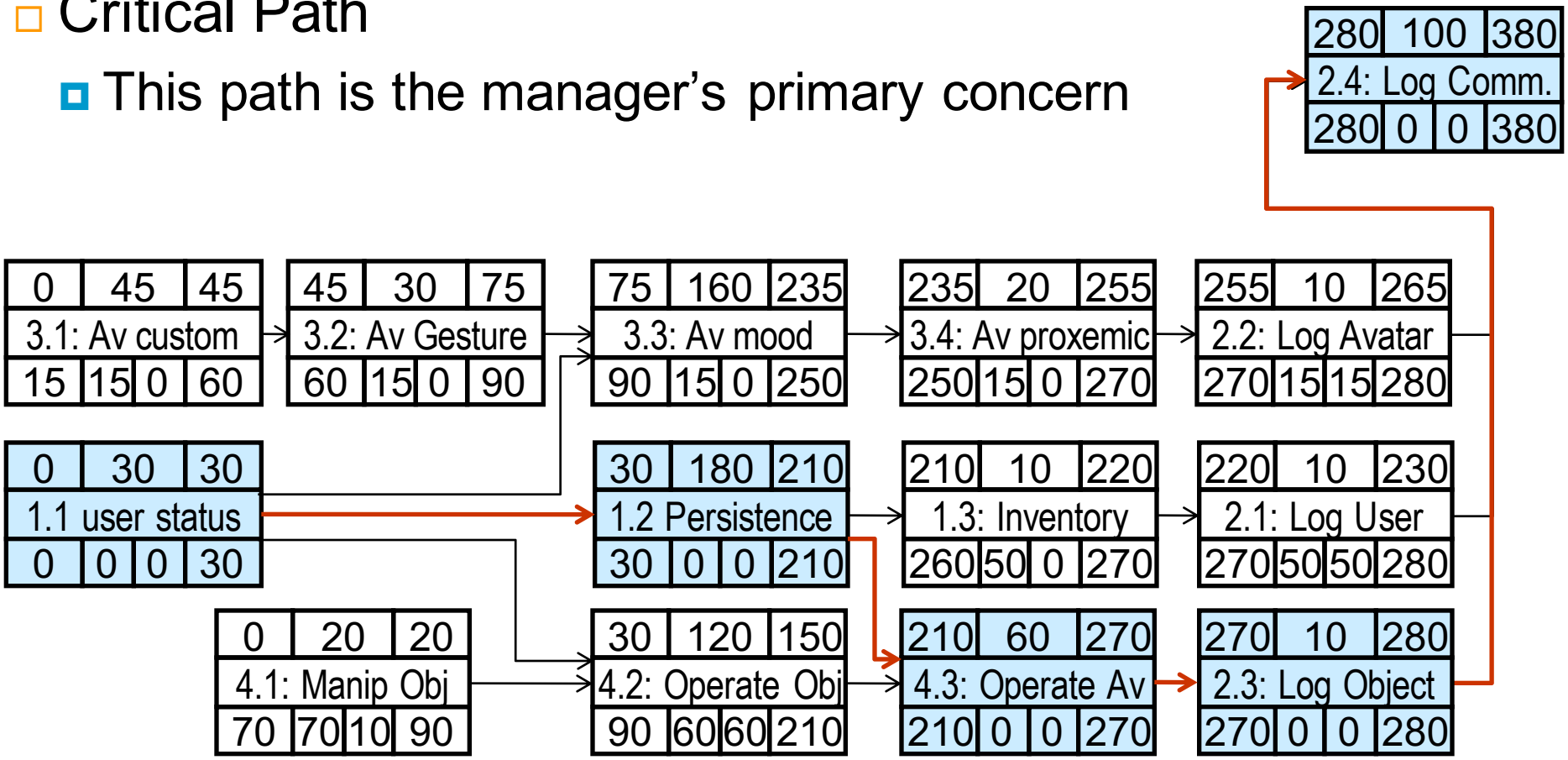
30	120	150
4.2: Operate Obj		
90	60	60	210

210	60	270
4.3: Operate Av		
210	0	0	270

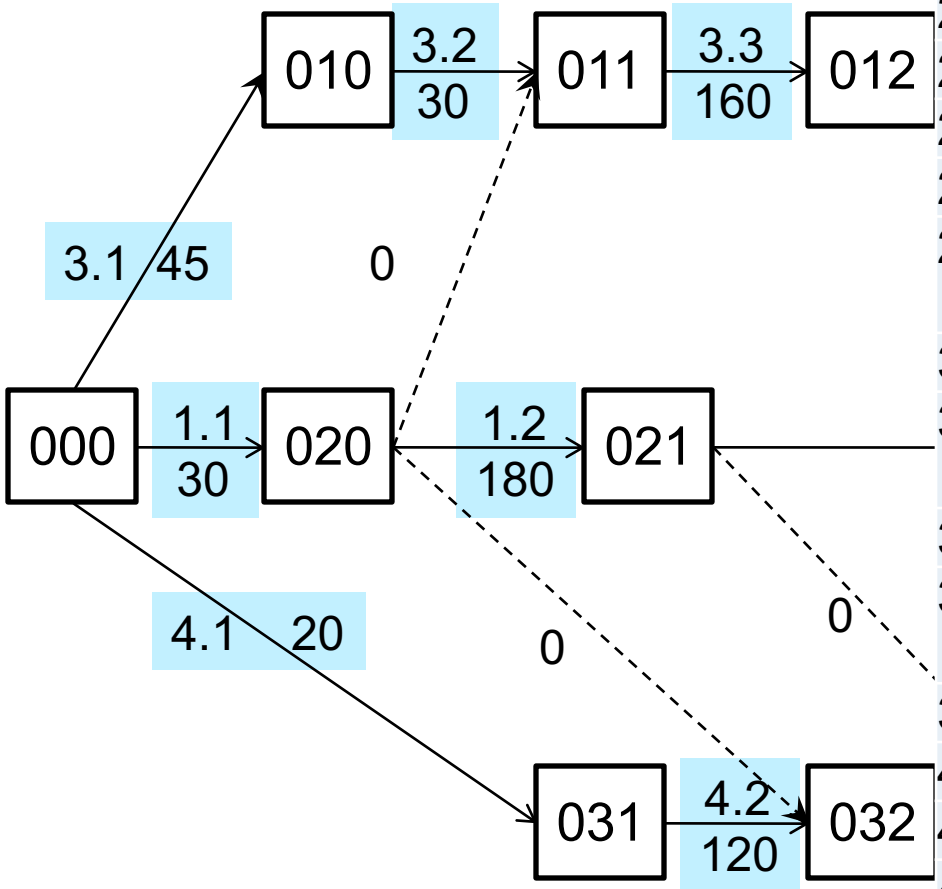
270	10	280
2.3: Log Object		
270	0	0	280

Analysis: Step #6

- Critical Path
  - This path is the manager's primary concern



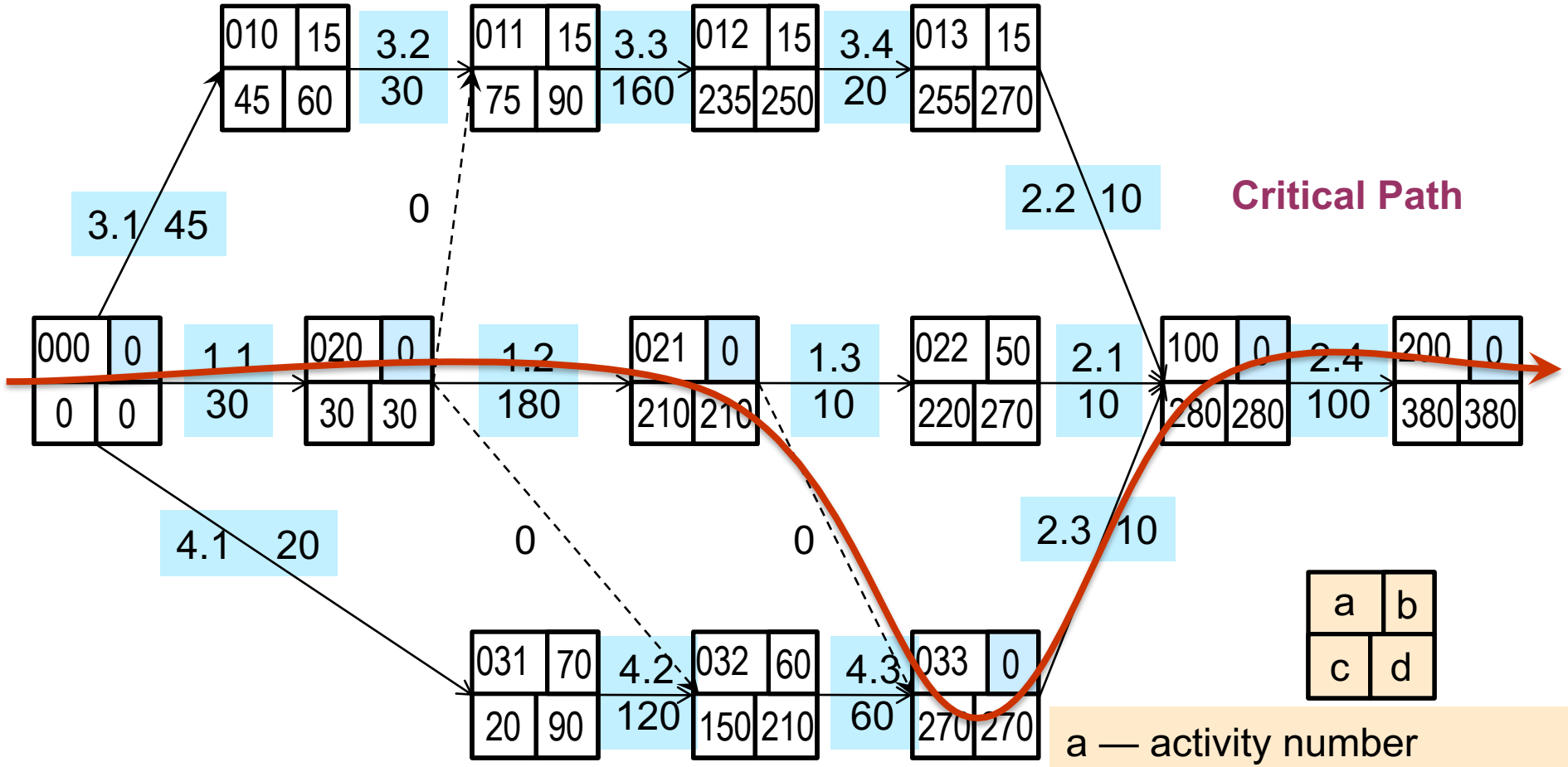
Example: Task Network (Milestone Network)



#	Activity	Len	Depend
1	Access		
1.1	User's status	30	
1.2	Persistence	180	1.1
1.3	Inventory	10	1.2
2	Logging		
2.1	Log user events	10	1.3
2.2	Log avatar events	10	3.4
2.3	Log object events	10	4.3
2.4	Log communication	100	2.1,2.2,2.3
3	Avatar		
3.1	Avatar customization	45	
3.2	Avatar Gesturing	30	3.1
3.3	Avatar mood indicator	160	3.2,1.1
3.4	Avatar proxemics	20	3.3
4	Interactions		
4.1	Manipulate objects	20	
4.2	Operate Objects	120	4.1,1.1
4.3	Operate on Avatar	60	1.2,4.2

4.3 Operate on Avatar

Example: Network Analysis



a — activity number
b — slack time
c — earliest completion time
d — latest completion time

Exercise: Network Analysis

- Do network analysis for a small development project

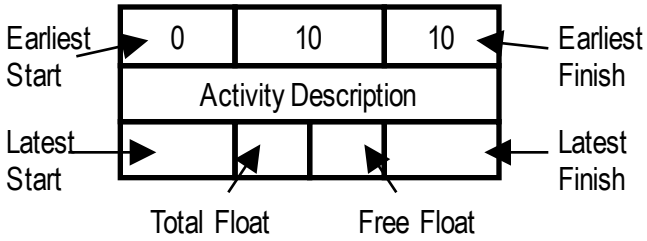
Task	Time	Dependencies
A: Analysis	6	
M: Modelling	8	A
I1: Implementation (Phase 1)	12	A
I2: Implementation (Phase 2)	8	I1
I3: Implementation (Phase 3)	14	M
UX: User Testing	5	I2, I3
D: Documentation	10	A
I: Installation and Customer Hand Over	10	D, UX

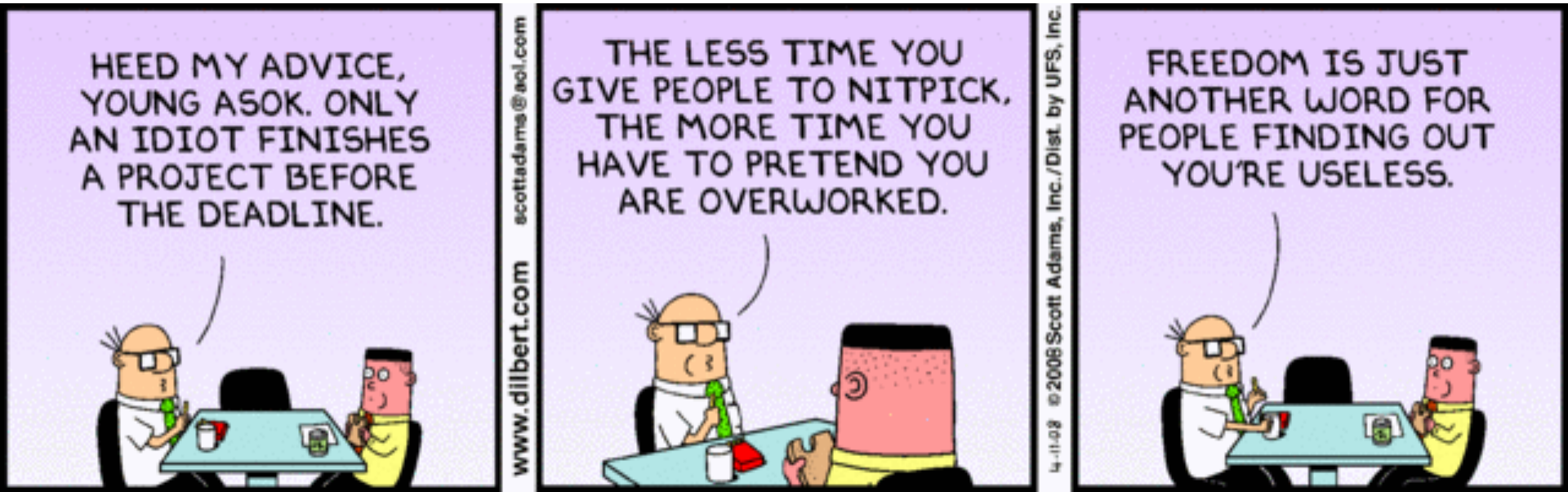
Solution:

Do it first!

See ASD-02-X-NetAnalEx.pdf

Password: bottom row of activity
I2: Implement 2
(as a six digit number)





Gantt Charts

Horizontal Bar Charts:

- Horizontal axis represents project time span
- Vertical axis represent project tasks

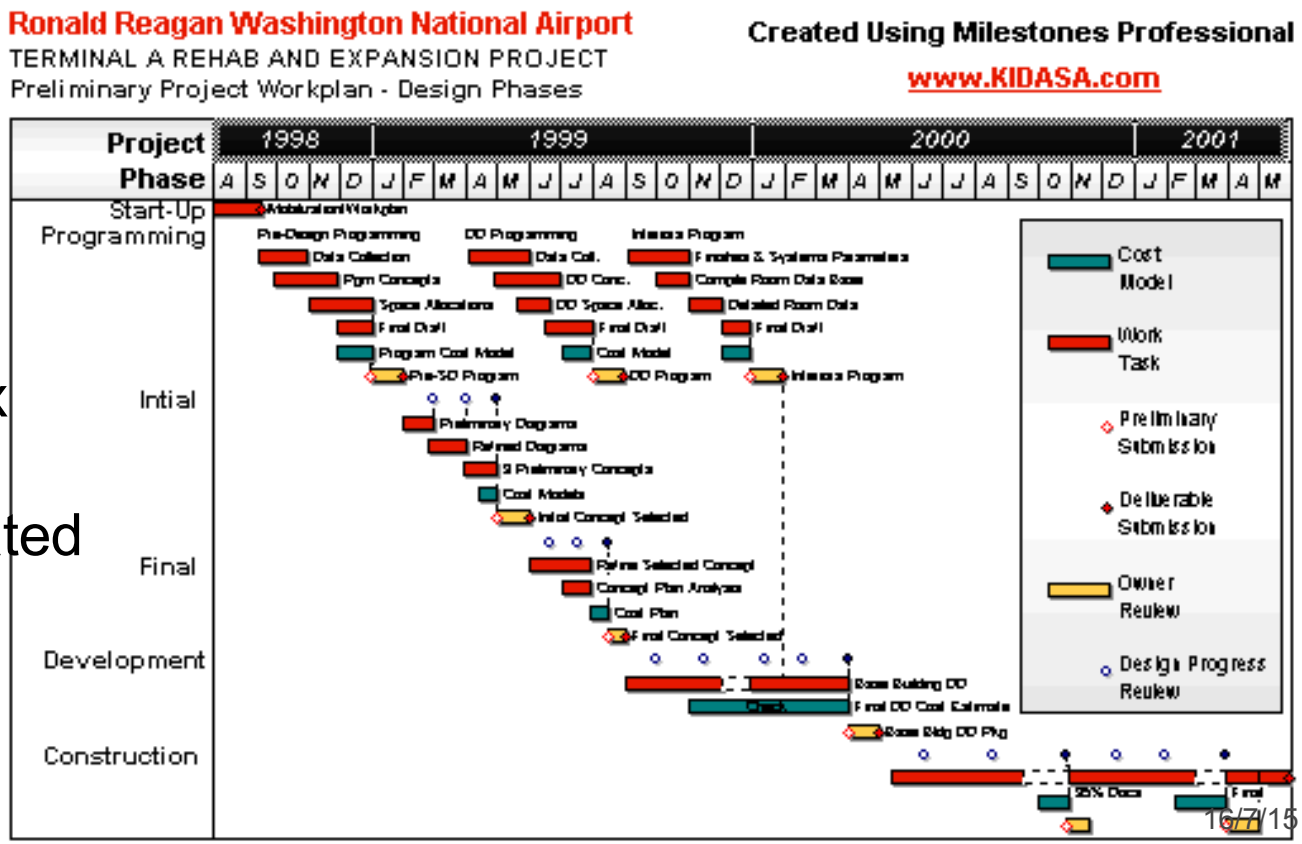
Capture:

- Task completion
- Simple dependencies
- Milestones and Deliverables

Can't handle complex task dependencies

Supported by automated scheduling tools

- Microsoft Project



Before Project

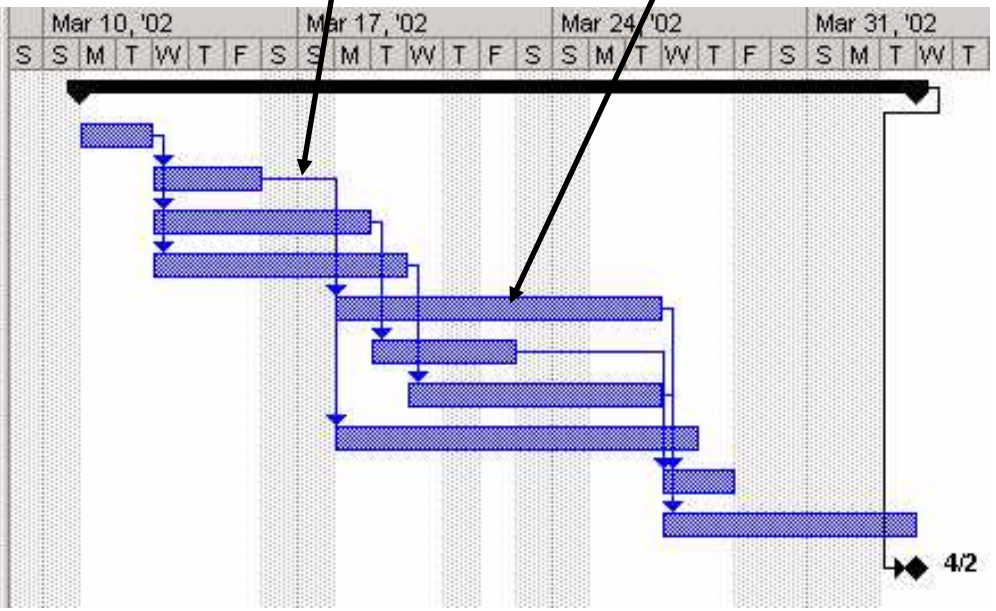
The timeline provides a chronological reference

Activity bars represent activity duration - the longer the bar, the longer the duration

Major project deliverables listed in bold and capped with a diamond

Arrows represent dependencies

	Task Nam	Duration	Start	Finish	Prede
1	**Proj**	**14 days**	**Mon 3/11/02**	**Tue 4/2/02**	
2	A	2 days	Mon 3/11/02	Tue 3/12/02	
3	B	3 days	Wed 3/13/02	Fri 3/15/02	2
4	C	4 days	Wed 3/13/02	Mon 3/18/02	2
5	D	5 days	Wed 3/13/02	Tue 3/19/02	2
6	E	6 days	Mon 3/18/02	Tue 3/26/02	3
7	F	3 days	Tue 3/19/02	Fri 3/22/02	4
8	G	4 days	Wed 3/20/02	Tue 3/26/02	5
9	H	7 days	Mon 3/18/02	Wed 3/27/02	3
10	I	2 days	Wed 3/27/02	Thu 3/28/02	6,7
11	J	3 days	Wed 3/27/02	Tue 4/2/02	8
12	Comple	0 days	Tue 4/2/02	Tue 4/2/02	1

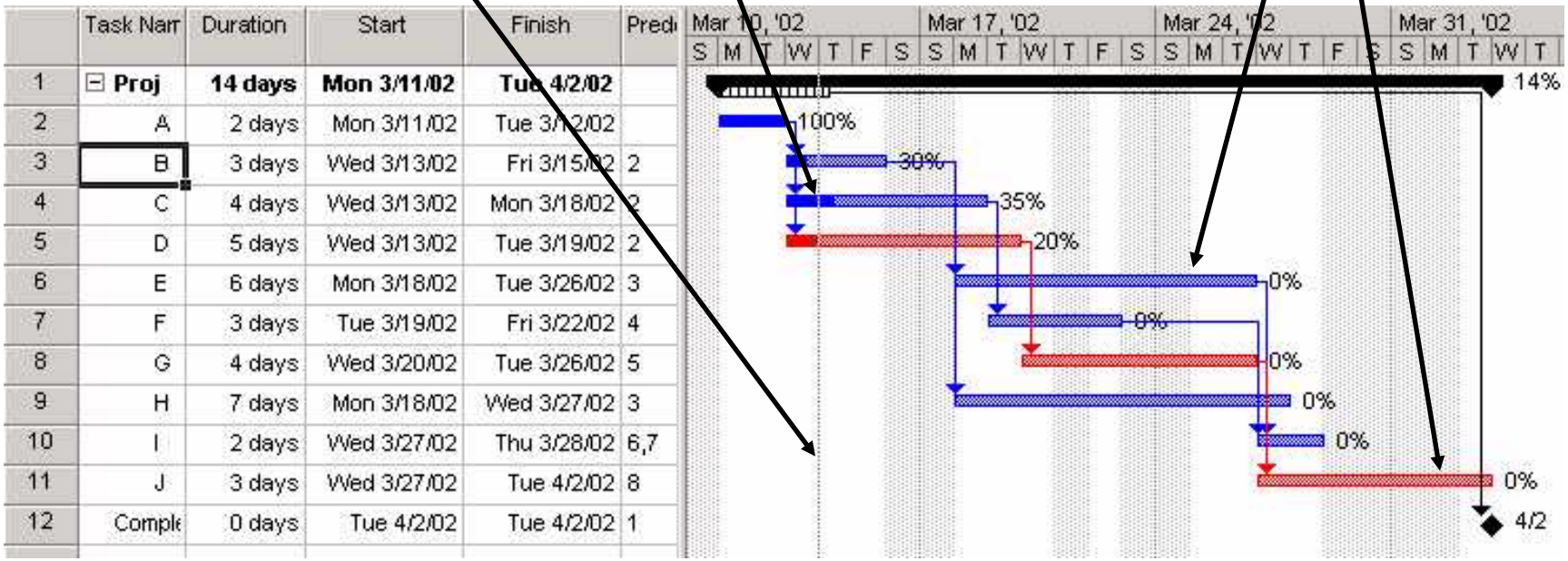


During Project

Current date represented by a dotted vertical line

Colour-coding assigns tasks to team members

Bars filled to show progress



	Task Nam	Duration	Start	Finish	Pred
1	**Proj**	**14 days**	**Mon 3/11/02**	**Tue 4/2/02**	
2	A	2 days	Mon 3/11/02	Tue 3/12/02	
3	B	3 days	Wed 3/13/02	Fri 3/15/02	2
4	C	4 days	Wed 3/13/02	Mon 3/18/02	2
5	D	5 days	Wed 3/13/02	Tue 3/19/02	2
6	E	6 days	Mon 3/18/02	Tue 3/26/02	3
7	F	3 days	Tue 3/19/02	Fri 3/22/02	4
8	G	4 days	Wed 3/20/02	Tue 3/26/02	5
9	H	7 days	Mon 3/18/02	Wed 3/27/02	3
10	I	2 days	Wed 3/27/02	Thu 3/28/02	6,7
11	J	3 days	Wed 3/27/02	Tue 4/2/02	8
12	Comple	0 days	Tue 4/2/02	Tue 4/2/02	1

RISK

Managing Risks

- Why?
  - Projects have a high level of uncertainty
  - Better to anticipate problems in advance
- How?
  - Identify specific risks to the project
  - Analyze the risks
  - Rank them in a particular order
  - Plan for monitoring, mitigation, management
  - Revisit continually during project

Boehm's top ten risk items + 2

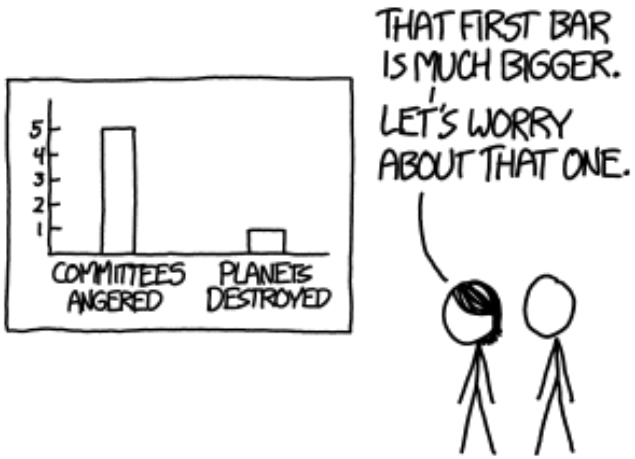
- Being blindsided by the competition
- Outside interruptions (e.g., ill-health)

1. Personnel shortfalls — Failure to recruit or retain key staff
2. Unrealistic schedules and budgets
3. Developing the wrong functions
4. Developing the wrong user interfaces
5. Gold-plating
6. Continuing stream of requirements changes
7. Shortfalls in externally-performed tasks
8. Shortfalls in externally-furnished components
9. Real-time performance shortfalls
10. Straining computer science capabilities

Risk Matrix

- Sort risk by a combination of:
  - Probability (high, medium, low)
  - Impact — catastrophic (project failure), critical (massive delay), marginal, negligible

		Probability		
		Low	Medium	High
Impact	Catastrophic	C	B	A
	Critical	D	C	B
	Marginal	E	D	C
	Negligible	F	E	D



What would happen if one tried to funnel Niagara Falls through a straw?
https://what-if.xkcd.com/147/

3M's

- Mitigation:
  - How can we avoid or reduce the risk?
    - avoid the risk: change requirements
    - transferring the risk: e.g., buy insurance
  - Or assume the risk and accept and control it
- Monitoring:
  - What factors can we track that will enable us to determine if the risk is becoming more or less likely?
- Management:
  - What contingency plans do we have if the risk becomes a reality

Examples of Risk I

Risk Condition	Consequence	Cat	Prob	Imp-act	Mitigation	Monitoring	Manage-ment
Competitors duplicate technology rapidly	Competitors products dilute market share	Follow-On	High	Critical	Develop Copy and IP protection, protect trade secrets through NDAs and source code through copyright	Keep abreast of current research and potential competitors	Adjust product pricing, develop upgrades
Product not well targeted to market	Unable to Capture Market Share	Commercialization	Medium	Critical	Create Business Plan and adapt development accordingly	Tracking current and predicting future market trends	Further market research and retargeting of product

Examples of Risk II

Risk Condition	Consequence	Cat	Prob	Imp-act	Mitigation	Monitoring	Management
Failure to recruit suitable lead developer on schedule	Significant delays (lead developer is on critical path)	Development	Medium	Critical	Advertise widely, offer attractive salary package	Track number and quality of applicants	Second round of recruiting (perhaps internationally), Target specific individual
Failure to acquire international business partner	International marketing and distribution hampered	Follow-On	Medium	Critical	Begin search early, develop business plan and prototypes to promote concept	Keep record of potential partners and commitment status	Delay follow-on until partner recruited or revise business strategy

Risk: Competition in Technology

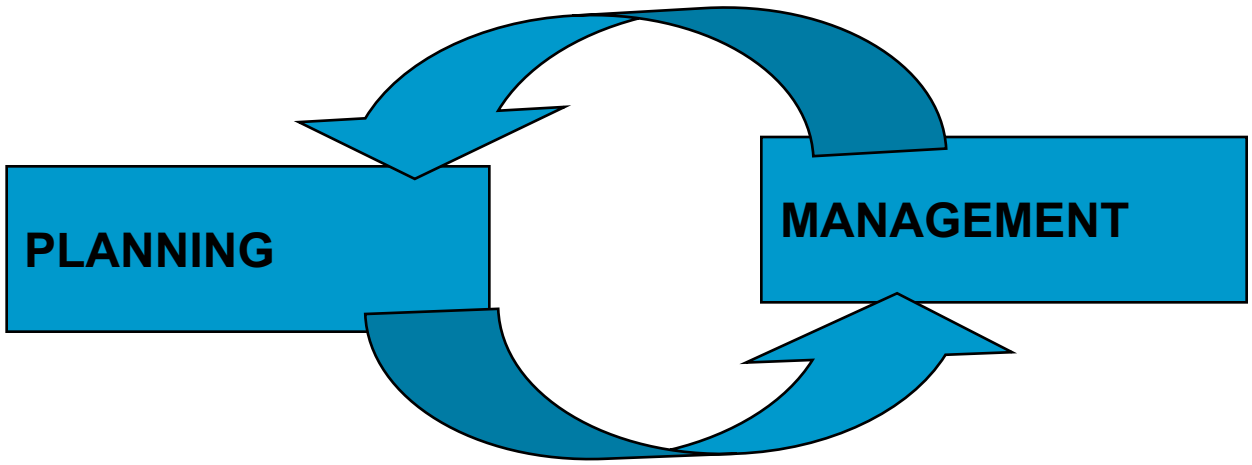
Risk Condition	Consequence	Mitigation	Monitoring	Management
Competitors duplicate technology rapidly	Competitors products dilute market share	Develop Copy and IP protection, protect trade secrets through NDAs and source code through copyright	Keep abreast of current research and potential competitors	Adjust product pricing, develop upgrades

Update 9/5/20..	Update 23/10/20..
Virtools and Prof. Kaufman identified as key competitors	3D Games Studio is a competitor, but not Prof. Kaufman

CONCLUSION

Planning vs. Management

- Planning
  - Pre- and post-
  - Network analysis, resourcing, risks, schedule
- Management
  - During
  - Controlling resources and timescales



Some Tips on Project Control

Remember to update planning documents

- Show progress in Gantt Chart
- Reassess risks

Resources

Bennett, McRobb & Farmer, *Object-Oriented Systems Analysis and Design: Using UML*, McGraw-Hill.

Chapter 22: Managing Object-Oriented Projects
highered.mcgraw-hill.com/sites/0077125363/student_view0/online_chapters.html

Sommerville, *Software Engineering*, Pearson

Chapter 22: Project management: Risk management, Managing people, Teamwork

Chapter 23: Project planning, Software pricing, Plan-driven development, Project scheduling, Agile planning, Estimation techniques