Introduction to Project Management

CS 146

Intro to Web Programming and Project Development

What is a project?

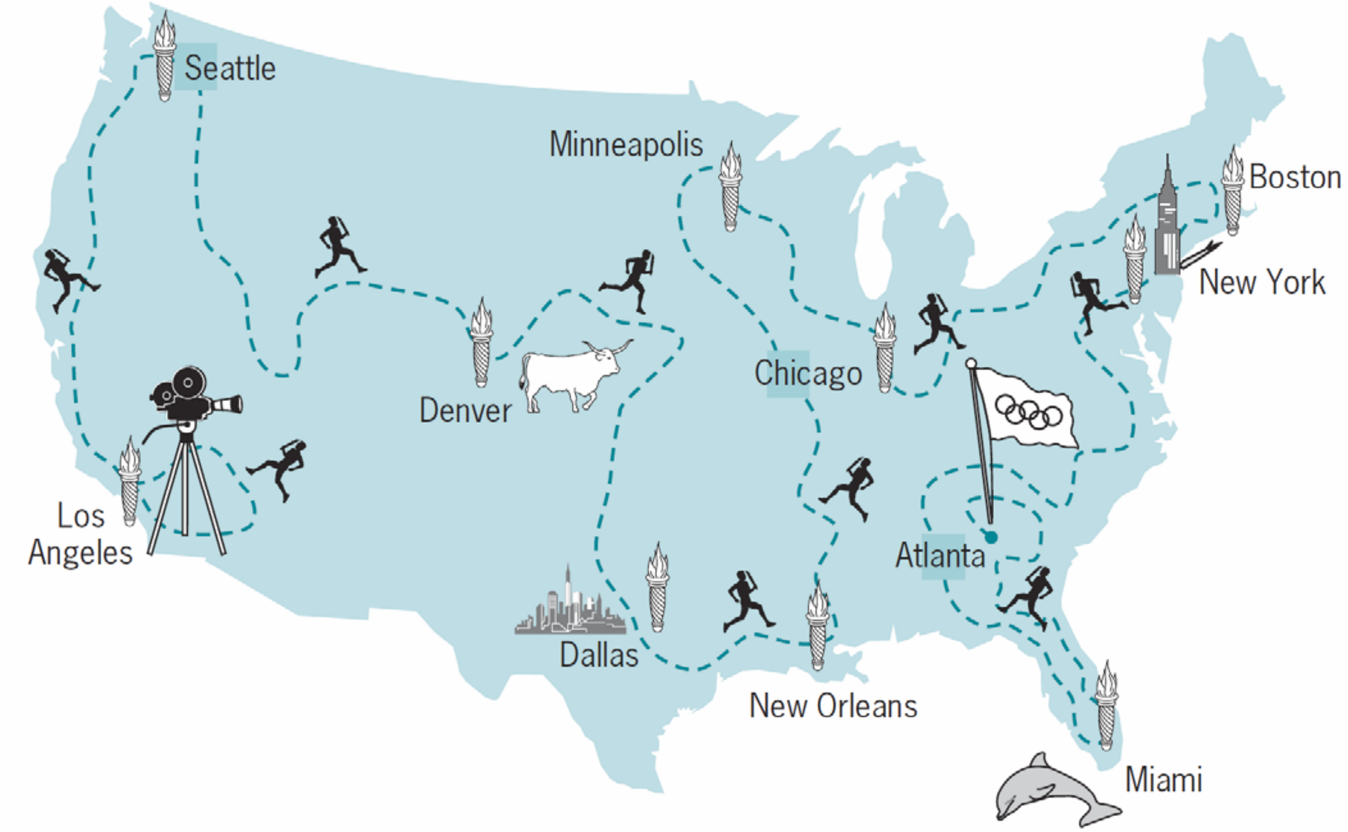
- The PMI (Project Management Institute) has defined a project as “A temporary endeavor undertaken to create a unique product or service.”
  - Building Egyptian pyramids
  - Constructing Boulder Dam
- Modern project management, however, is usually said to have begun with the Manhattan Project.

Other Similar Terms

- …Created by the military
  - *program* refer to an exceptionally large, long range objective that is broken down into a set of *projects*.
  - *projects* are divided further into *tasks*.
  - *tasks* are, in turn, split into *work packages* that are themselves composed of *work units*.
- Summary: a *project* is a **specific, finite task** to be accomplished.

Project Management in Practice

1996 Atlanta Olympic Torch Relay



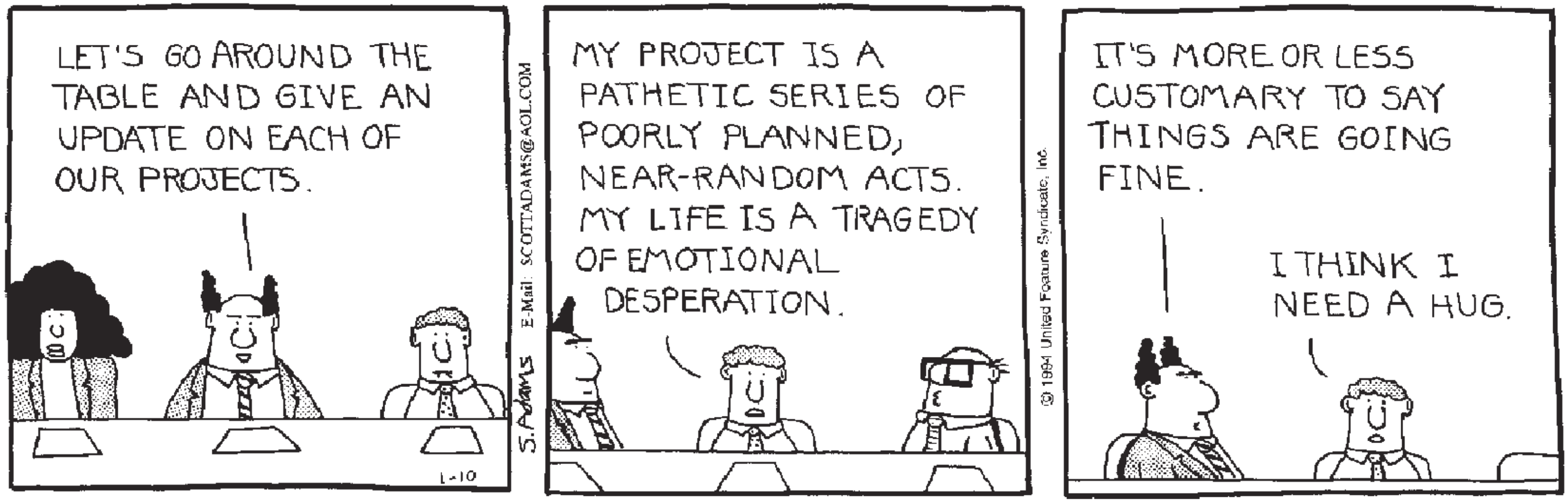
Project Management in Practice

- Planning for the 1996 Atlanta Olympic Torch Relay
  - took two years,
  - cost over $20 million
  - involved an 84 day, 42 state campaign using 10,000 runners to carry the torch for 15,000 miles!
  - included a 40-vehicle caravan carrying security officers, media personnel, medical personnel, computers, telecommunications gear, clothing, food, and spare lanterns with extra flames in case the original torch went out.
  - The caravan included: 50 cellular telephones; 60 pagers; 120 radios; 30 cars; 10 motorcycles; and clothing for 10,000 runners, 10,000 volunteers, as well as 2,500 escort runners

Project Manager (PM)

- Often lacks authority a level consistent with his or her responsibility.
- Is expected to coordinate and integrate all activities needed to reach the project’s goals.
- The project form of organization allows the manager to be responsive to:
  - the client and the environment
  - Identify and correct problems at an early date
  - make timely decisions about trade-offs between conflicting project goals
  - ensure that managers of the separate tasks do not optimize the performance of their tasks at the expense of the total project

A little humor…

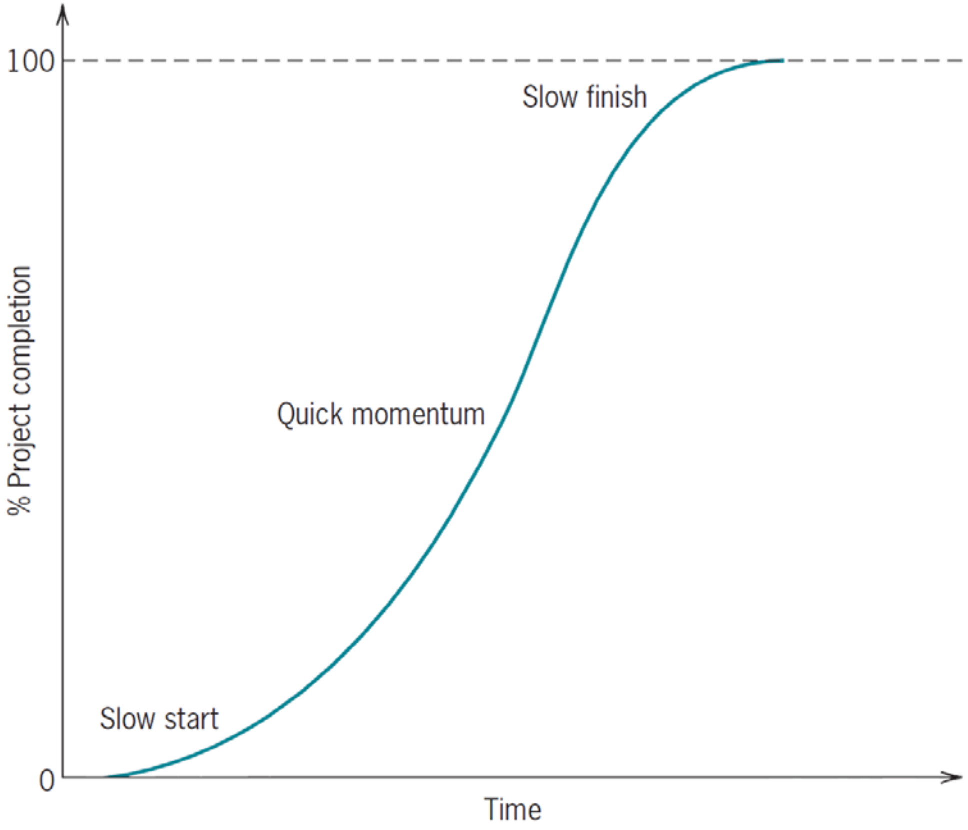

LET'S GO AROUND THE TABLE AND GIVE AN UPDATE ON EACH OF OUR PROJECTS.
1-10
S.Adams E-Mail: SCOTTADAMS@AOL.COM
MY PROJECT IS A PATHETIC SERIES OF POORLY PLANNED, NEAR-RANDOM ACTS. MY LIFE IS A TRAGEDY OF EMOTIONAL DESPERATION.
© 1994 United Feature Syndicate, Inc.
IT'S MORE OR LESS CUSTOMARY TO SAY THINGS ARE GOING FINE.
I THINK I NEED A HUG.

DILBERT: © Scott Adams/Dist. by United Feature Syndicate, Inc.

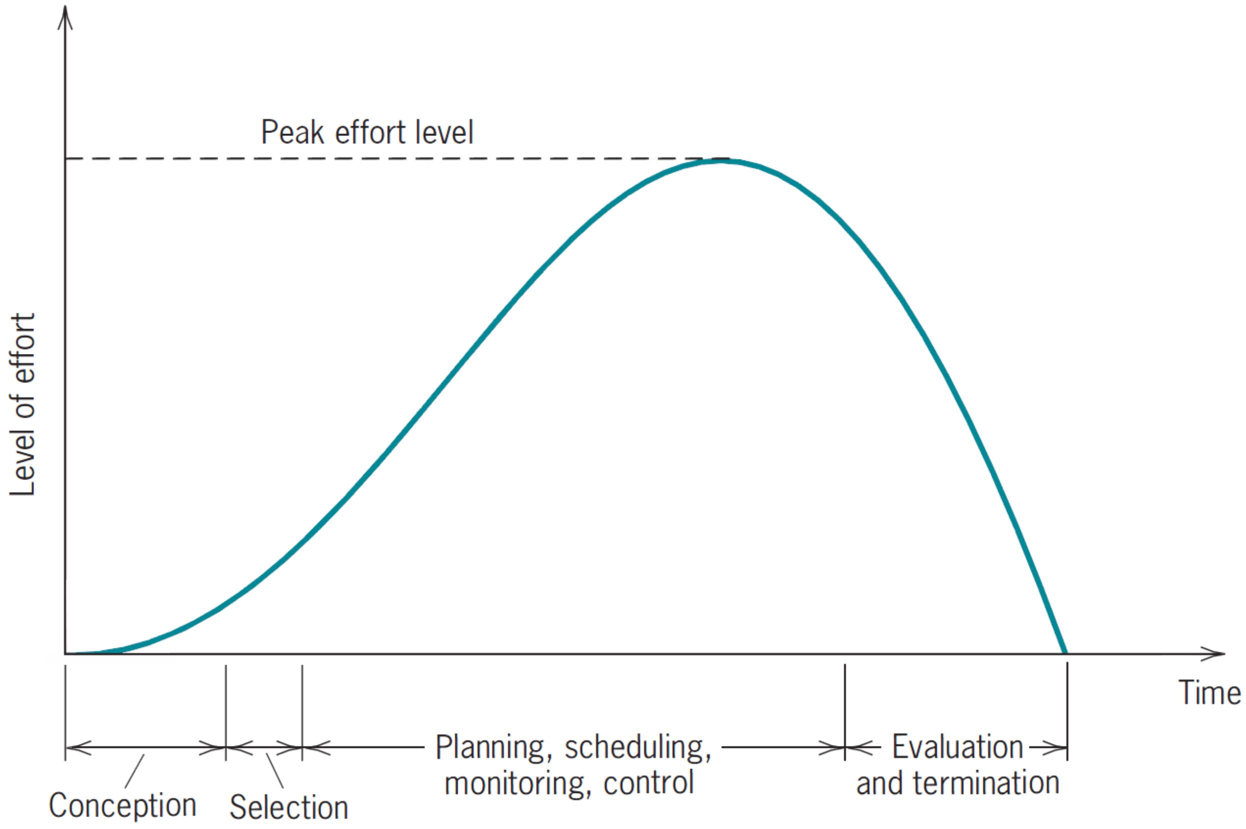
Project Lifecycle

- There are many models, but we'll see the most basic one: the waterfall model.
  - Each step is completed before heading to the next.
- Typical steps in CS projects include: Inception, Requirements, Analysis, Design, Implementation, Testing, Debugging, Production, Distribution, and Maintenance
  - We will talk of all of these in detail.
- The PM is responsible for organizing these steps properly and planning the schedule, budget, and projected performance.

Project Lifecycle

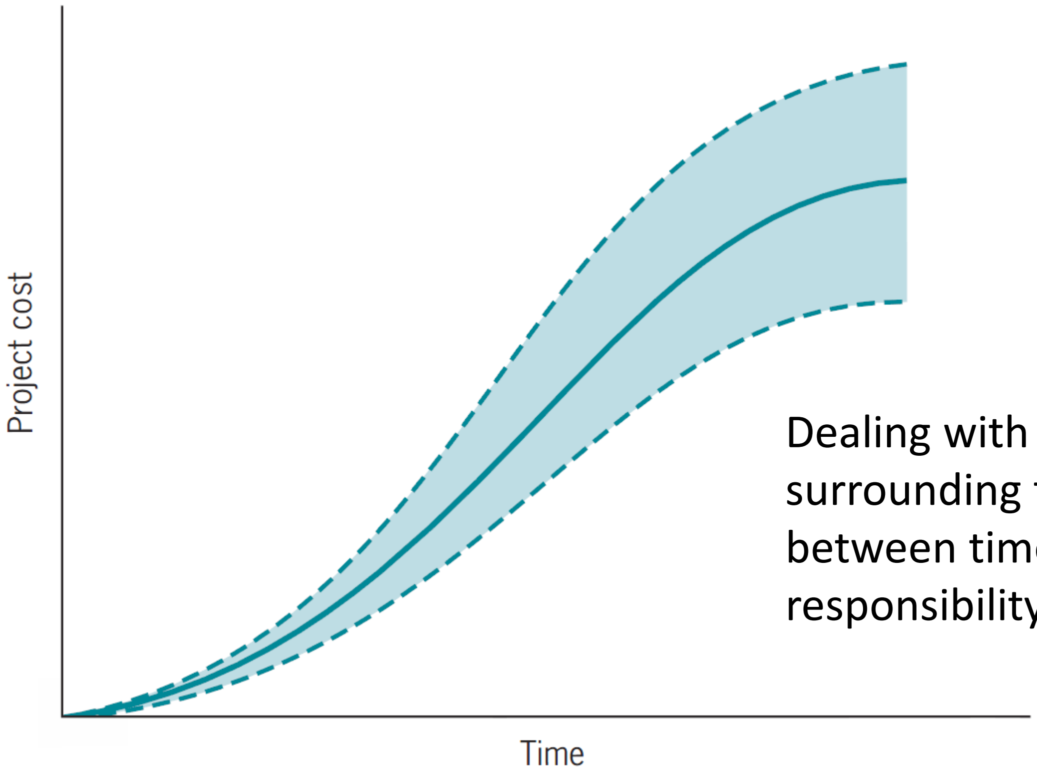


Time Distribution of Project Effort



Estimate of Project Cost:

Estimate made at the start



Dealing with the uncertainty surrounding the relationship between time and cost is a major responsibility of the PM.

Classroom Discussion

- Describe some situations in which project management would probably not be effective.
- How might project management be used when doing a major schoolwork assignment?
- Would you like to be a project manager? Why, or why not?

The Vital Dozen for Project Managers

- “What information were you never given as a novice project manager that, in retrospect, could have made your job easier?”
- The Vital Dozen is the result of interviews conducted with dozens of senior project managers that were asked the simple question above.

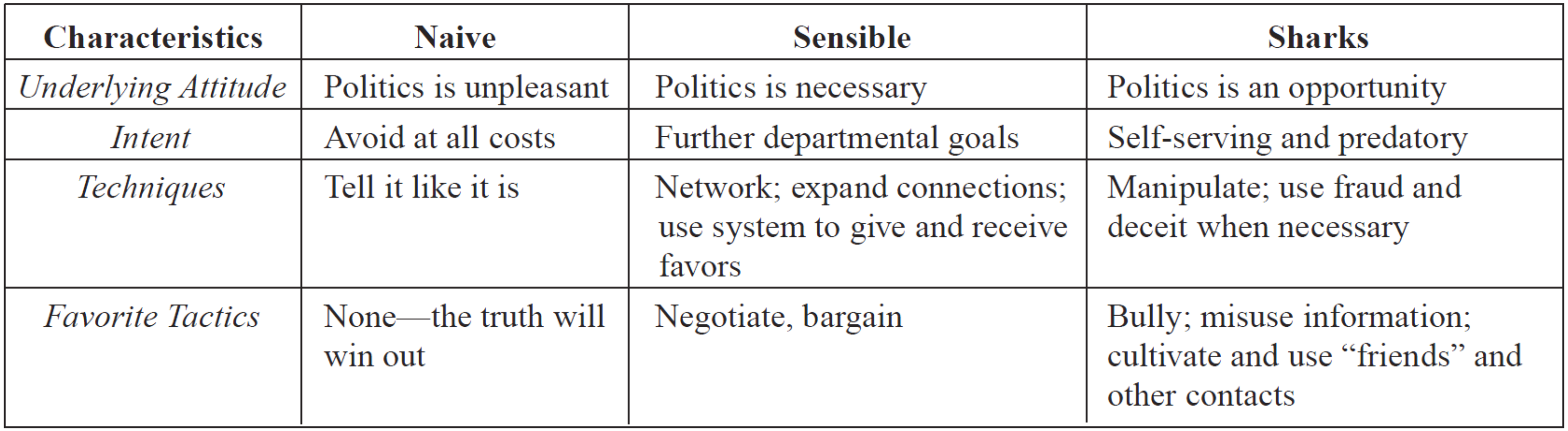
The Vital Dozen for Project Managers

1. **Understand** the context of project management.
2. **Recognize** project team conflict as progress.
3. **Understand** who the stakeholders are and what they want.
4. **Accept** and use the political nature of organizations.
5. **Lead** from the front.
6. **Understand** what "success" means.

The Vital Dozen for Project Managers

7. **Build** and maintain a cohesive team.
8. **Enthusiasm** and despair are both infectious.
9. **One look** forward is worth two looks back.
10. **Remember** what you are trying to do.
11. **Use time** carefully or it will use you.
12. **Above all,** plan, plan, plan.

Characteristics of Political Behaviors

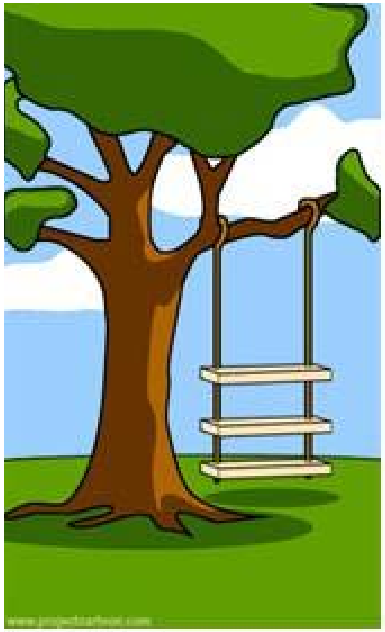


Characteristics	**Naive**	**Sensible**	**Sharks**
Underlying Attitude	Politics is unpleasant	Politics is necessary	Politics is an opportunity
Intent	Avoid at all costs	Further departmental goals	Self-serving and predatory
Techniques	Tell it like it is	Network; expand connections; use system to give and receive favors	Manipulate; use fraud and deceit when necessary
Favorite Tactics	None—the truth will win out	Negotiate, bargain	Bully; misuse information; cultivate and use "friends" and other contacts

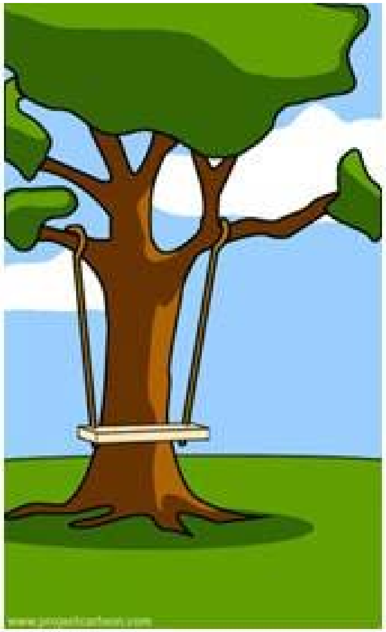
Six Practical Suggestions

- Six practical suggestions to help project managers control their tasks and projects without feeling constantly behind schedule:
  - Create a realistic time estimate without overextending yourself.
  - Be absolutely clear about what the boss or client requires.
  - Provide for contingencies (schedule slippage, loss of key team member).
  - Revise original time estimate and provide a set of options as required.
  - Be clear about factors that are fixed (specifications, resources, and so on).
  - Learn to say “Yes, and . . .” rather than “No, but . . .” Negotiation is the key.

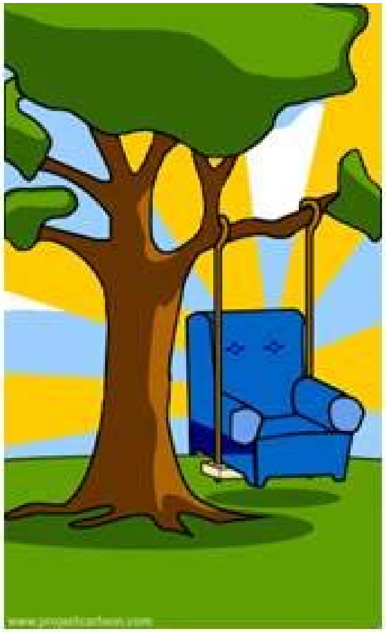
Legendary Tree Swing: Foolishness in Managing Software Projects



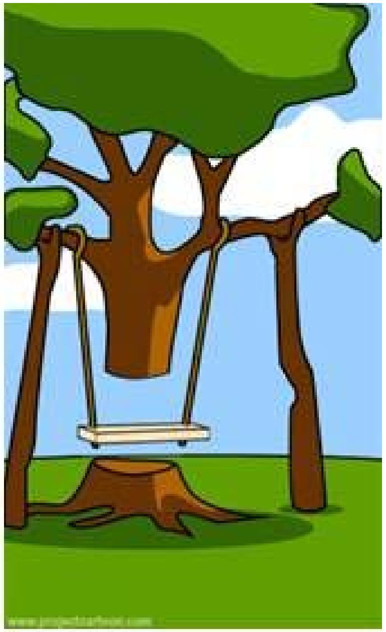
How the customer explained it



How the Project Leader understood it

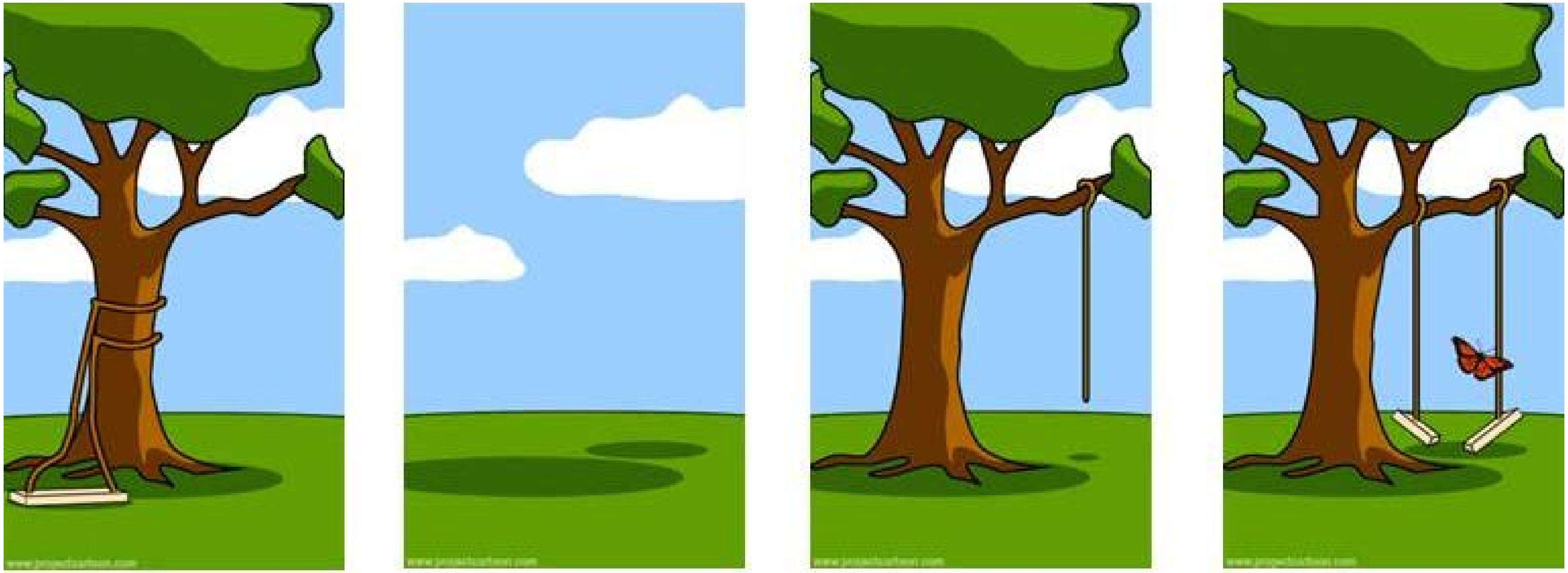


How the Business Consultant described it



How the Analyst designed it

Legendary Tree Swing: Foolishness in Managing Software Projects



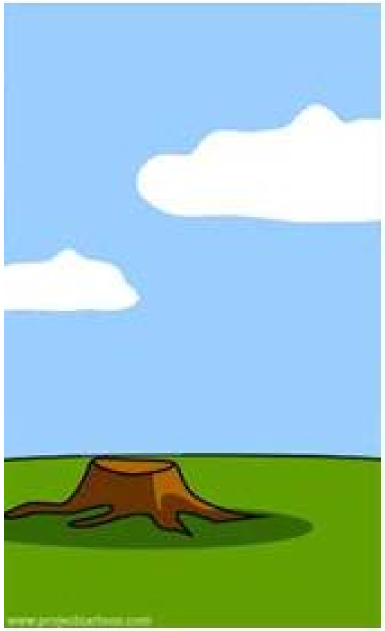
How the Programmer wrote it

How the project was documented

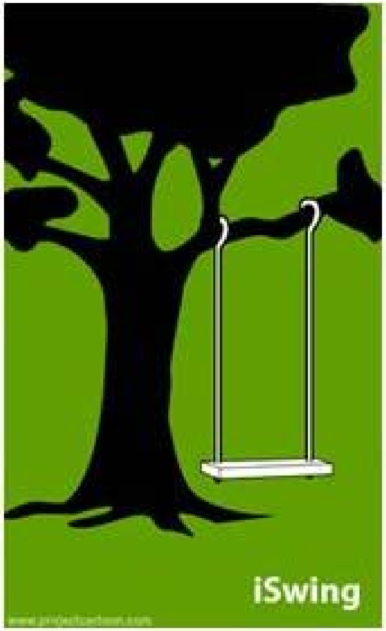
What Operations installed

How it performed under load

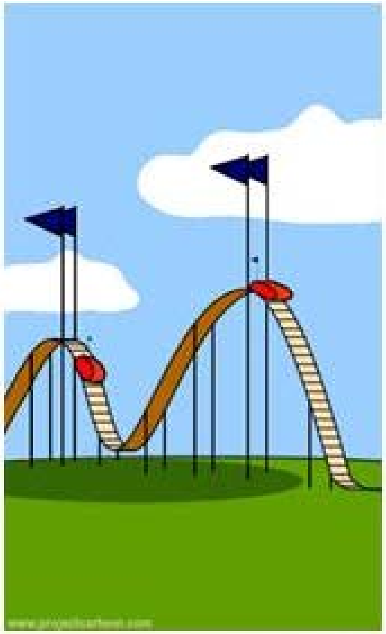
Legendary Tree Swing: Foolishness in Managing Software Projects



How it was supported



What marketing advertised



How the customer was billed

www.projectcartoon.com

What the customer really needed

Inception / Project Selection

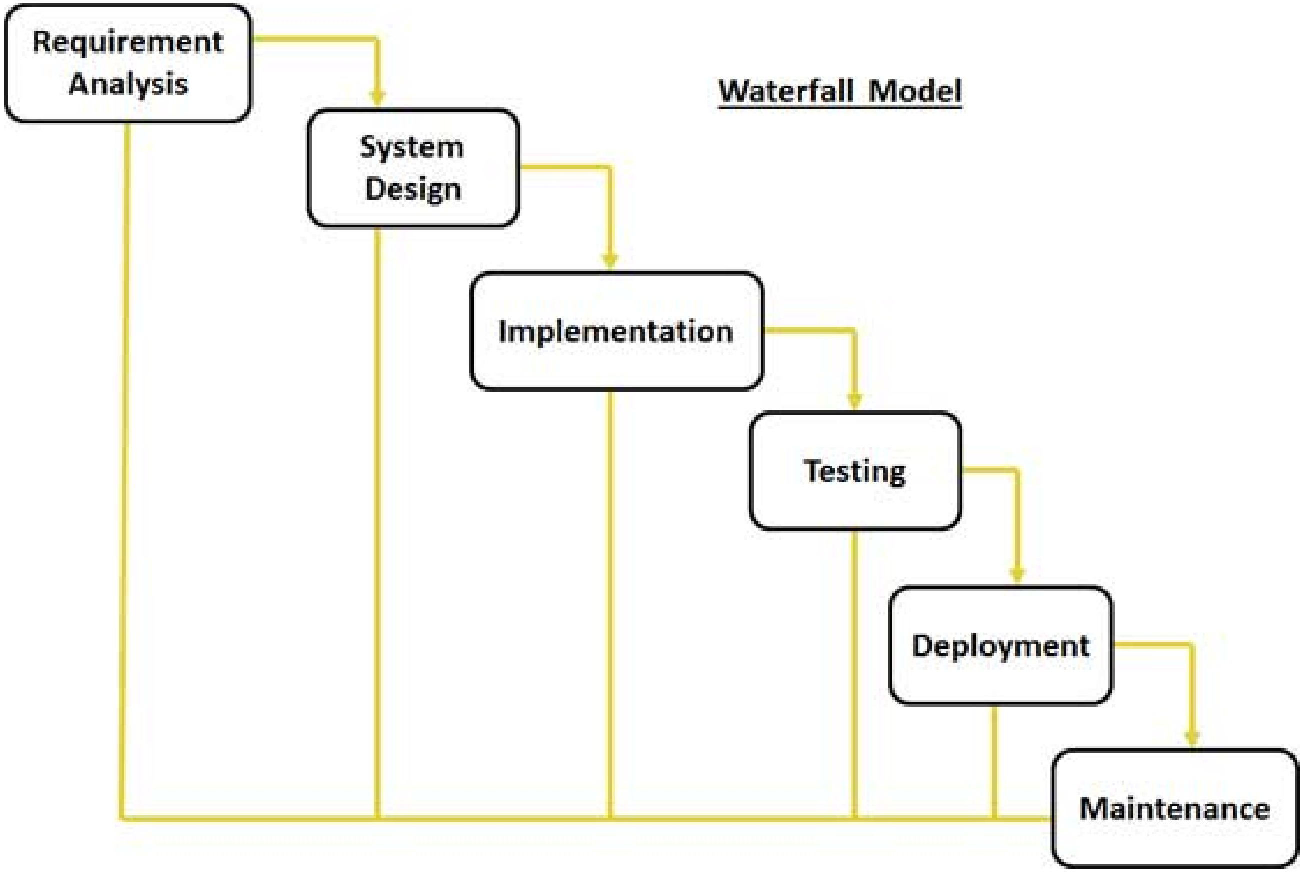
- All projects start from an idea, and often multiple ideas will appear at the same time in a big company.
- Selecting the right project is the first task of a good project manager, so how does one choose?
  - First identify the type of projects in front of you and make a decision based on what's better for your company.
  - There are many selection models, but you should take a PM course to learn more about them.

SDLC: Waterfall Model

- The Waterfall Model was first Process Model to be introduced.
- It is also referred to as a linear-sequential life cycle model.
- It is very simple to understand and use.
- In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

SDLC = Software Development Life Cycle

Waterfall Model Visualized



Phases of Waterfall Model

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

When to Use the Waterfall Model

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Pros of Waterfall Model

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Cons of Waterfall Model

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- No working software is produced until late in the life cycle.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Beginning Requirement Analysis

- Before requirement development begins, identify all the stakeholders for the project.
  - Understand how stakeholders will be affected by the project’s outcome.
  - Understand what stakeholder’s involvement or influence will be in the project.
- Without this planning you may miss key stakeholders, requiring additional sessions to get their inputs.

Elicitation

- Observation
- Interview with stakeholders
- Existing documentation
- JAD sessions
  - Joint Application Design
  - Workshop where knowledge workers, IT specialists, and management meet to define and review business requirements
- User stories
- Use cases

Acceptance - Definition of Done

All stakeholders must have a common understanding of “done”.

It's my project...and It's not done until I say I'm done.
P.O
BZZZZT!
I'd say he's done. Let's move on...
By Clark & Vizdos
© 2006 implementingscrum.com

Basic definition of “done” means the requirement is designed, built, tested, and accepted by stakeholder.

Characteristics of Good Requirements

- Complete
  - Requirements should be as complete as possible—no open-ended requirements.
- Testable
  - Must be able to create a test for all requirements.
- Consistent
  - Requirements must be consistent with each other—no conflicts.
- Design Free
  - Software requirements should be specified in the business perspective rather than the software perspective.
- Unambiguous
  - Use "shall" and other related words. Don't be wishy-washy.

Will, Must, Shall, Should

Verb	Definition	Example
Shall	Indicative of a requirement	A report shall be needed to list everyone in the group with blue eyes.
Must	Critical Requirement Constraint	There must be a new eligibility interface designed for the new drug vendor.
Will	Declaration of fact	The group termination reports SR9999 will be changed to reflect the new term date format.
Should	Goal	Use of patch 334555 should improve performance of batch billing 50%.

Requirements Document - Summary

- A good requirements document is a simple, complete, and well-structured.
- 4 key points must be addressed
  - Description of the problem
  - Basic approach of solution
  - Constraints (time, budget, etc.)
  - How success will be measured (how do we know we're done?)