



 POLITECNICO DI MILANO

Dipartimento di  
Elettronica e Informazione

Planning and Managing Software Projects 2014-15  
Class 7

# Planning Phase – Part II

Matching Life Cycles to Project and Project Plans

**Emanuele Della Valle**

<http://emanueledellavalle.org>

- 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

- Last Class
  - Phases in Detail
    - Step-by-step of typical software project
  - Lifecycles
- Today
  - Matching Lifecycles to Project
  - Project plans
- Next Weeks:
  - Lots of Project-ish Details: WBS, PERT, CPM, Scheduling & Estimation

- Phases in Detail
  - Know your pure waterfall, 7 phase model
  - Understand the steps in each phase
  - Know typical deliverables of each
  - Know the primary issues and goals of each
  
- Lifecycles
  - Know a representative sample
  - Waterfall and variation, 1-2 iterative ones
  - Learn a bit about XP and other Agile methods

- a.k.a. Lifecycle Management or Systems Development Life Cycle (SDLC )
- Greatly influences your chance of success
- Not choosing a lifecycle is a bad option
- Three primary lifecycle model components
  - Phases and their order
  - Intermediate products of each phase
  - Reviews used in each phase

- Different projects require different approaches
- You do not need to know all models by name
- You should know how that if given a certain scenario what sort of SDLC would be appropriate
- There are more than covered here
- A lifecycle is not a design, modeling or diagramming technique
  - The same technique (UML, DFD, etc) can be used with multiple lifecycles

- Varies by project
- Opt for “iterative” or “incremental”
- How well are requirements understood?
- What are the risks?
- Is there a fixed deadline?
- How experienced is the team or customer?

Model	a	b	c	d	e	f	g	h	i	j	k
Pure Waterfall			x	x		~				~	~
Code-and-Fix							x				x
Spiral	x	x	x	x	x	~	~	~	x	x	
Modified Waterfalls (sashimi)	~	~	x	x	~	~	x	~	~	~	
Evolutionary Prototyping	x		~	x	~		~	x	x	~	
Staged Delivery			x	x	~	~	~		~	x	
Commercial (COTS)	x					x	x				~

Legend:  
 x = excellent,  
 ~ = fair to excellent,  
 empty box = poor.

- a. Works with poorly understood requirements
- b. Works with poorly understood architecture
- c. Produces highly reliable system
- d. Produces system with large growth envelope
- e. Manages risks
- f. Can be constrained to a predefined schedule
- g. Has low overhead
- h. Allows for midcourse corrections
- i. Provides customer with progress visibility
- j. Provides management with progress visibility
- k. Requires little manager or developer sophistication



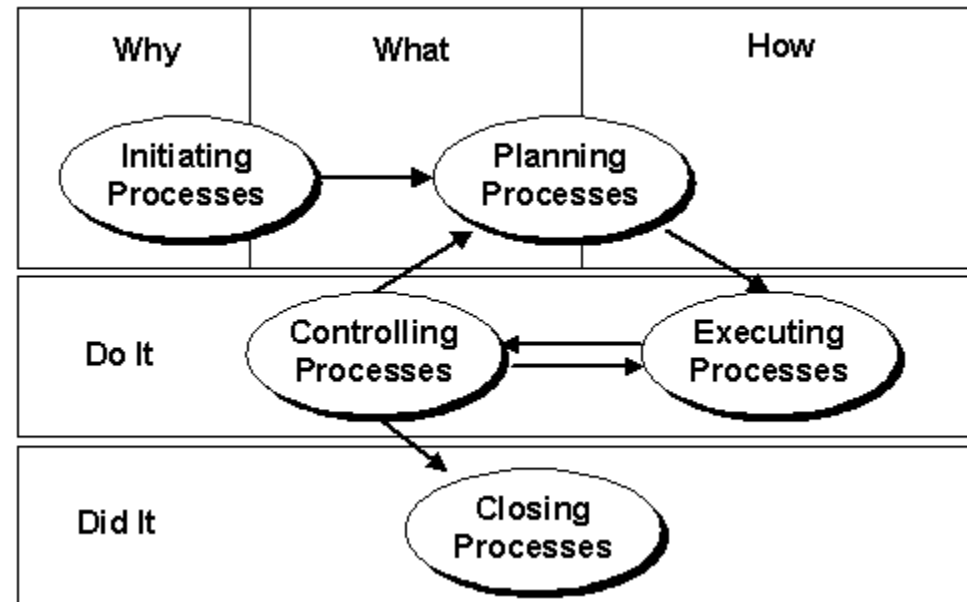
Model	a	b	c	d	e	f	g	h	i	j	k
Pure Waterfall			x	x		~				~	~
Code-and-Fix							x				x
Spiral	x	x	x	x	x	~	~	~	x	x	
Modified Waterfalls (sashimi)	~	~	x	x	~	~	x	~	~	~	
Evolutionary Prototyping	x		~	x	~		~	x	x	~	
Staged Delivery			x	x	~	~	~		~	x	
Commercial (COTS)	x					x	x				~
Rapid Application Development	x	x		x	~	x	~	x	x	x	
Extreme Programming	x	x		x	~		x	x	x	~	
Agile Software Development	x	x		x	~		~	x	x	~	
SCRUM	x	x		x	~	x	~	x	x	~	

Legend:  
 x = excellent,  
 ~ = fair to excellent,  
 empty box = poor.

- “Plans are nothing. But planning is everything.”  
-- Gen. Dwight Eisenhower

- Preliminary planning starts on day one
- Even in the pre-project phase
- Should not be conducted “in secret”
- Need buy-in and approval
  - Very important step
  - Both from above and below

- Why
  - Deliverable: ROI
- What
  - SOW, Requirements
- How
  - Design Specification, Software Development Plan, Lifecycle
- Do it
  - Execution
- Did it
  - Post Project Report



Futrell, Shafer, Shafer, "Quality Software Project Management"

# Primary Planning Steps

- Identify project scope and objectives
- Identify project organizational environment
- Analyze project characteristics
- Identify project products and activities
- Estimate effort for each activity
- Identify risk
- Allocate resources
- Review and communicate plan

- Planning
- Product

- Software Development Plan (SDP)
- Software Quality Assurance Plan (SQAP)
- Software Configuration Management Plan (SCMP)
- Risk Management Plan
- Software Process Improvement Plan
- Communications Management Plan
- Migration Plan
- Operations Plan

- You (the PM) need to choose which documents are appropriate
- Docs do not have to be lengthy
- Small Set:
  - Software Development Plan
  - Risk Management Plan
  - Software Quality Assurance Plan
  - Software Configuration Management Plan



- Project Charter
- Statement of Work (SOW)
- Software Project Management Plan (SPMP)
- Budget
- Responsibility Assignment Matrix (RAM)
- Risk Management Plan

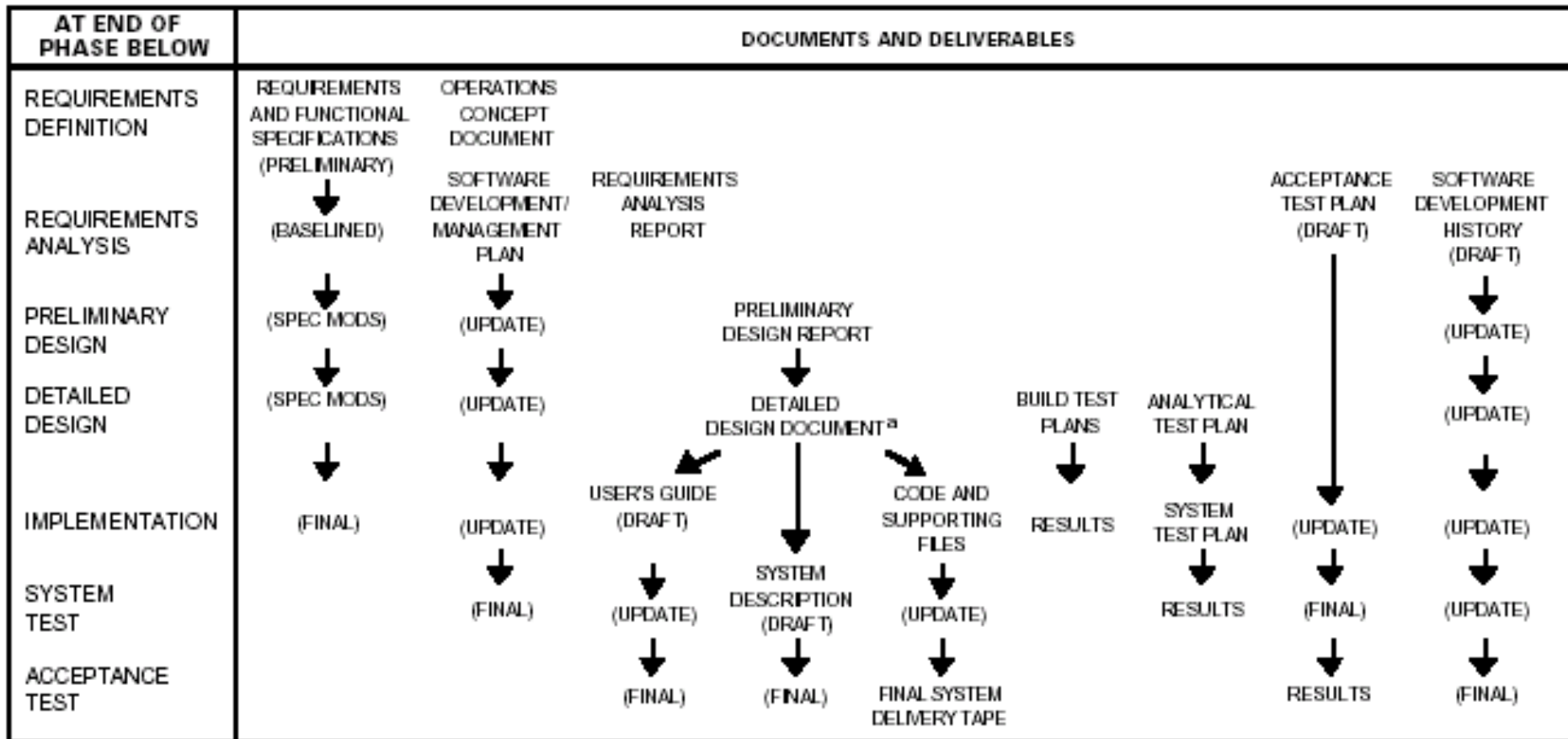
- Statement of Need
- System Interface Specification
- Software Requirements Specification
- Software Design Specification
- Software Validation & Verification Plan
- User Documentation
- Support Plan
- Maintenance Documentation

- Another McConnell book
- See construx.com's SPSG section  
[http://www.construx.com/thought\\_leadership/books/survival\\_guide/](http://www.construx.com/thought_leadership/books/survival_guide/)
  - Good content online
  - Documents
  - Schedules
  - Checklists
  - Project web site template
- I tool I've often used
  - Software Project Survival Test
    - [http://www.construx.com/Thought\\_Leadership/Books/Survival\\_Guide/Resources\\_by\\_Subject/Survival\\_Test/](http://www.construx.com/Thought_Leadership/Books/Survival_Guide/Resources_by_Subject/Survival_Test/)

- How much will it cost?
- How long will it take?
- How many people will it take?
- What might go wrong?

- Scoping
- Estimation
- Risk
- Schedule
- Control Strategy

- You want a fairly sophisticated process without incurring much overhead
- Remember, projects are often larger than they first appear
- Easier to loosen too much process than add later



NASA's "Manager's Handbook for Software Development"

- Software Project Management Plan (SPMP)
- Some consider it the most important document in the project (along with SRS)
  - Can be seen as an aggregation of other core documents
- Evolves over time as pieces come together



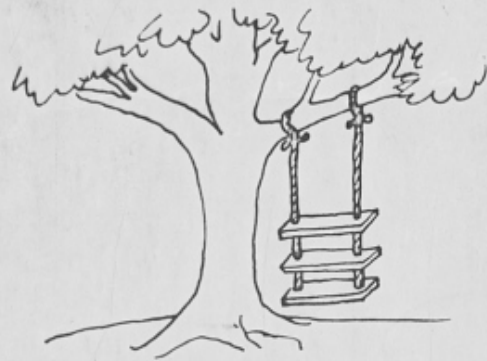
- Fundamental Sections
  - Project overview
  - Deliverables
  - Project organization
  - Managerial processes
  - Technical processes
  - Budget
  - Schedule

- Often a section of SPMP
- Describes information flow to all parties
  - Gathering and distributing information
- Status meetings
  - Monthly, Weekly, Daily?
  - Status reports are vital

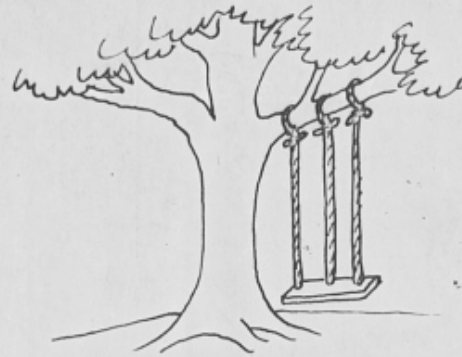
# Create a Project Intranet

- A great communications tool
- Reference all project resources here

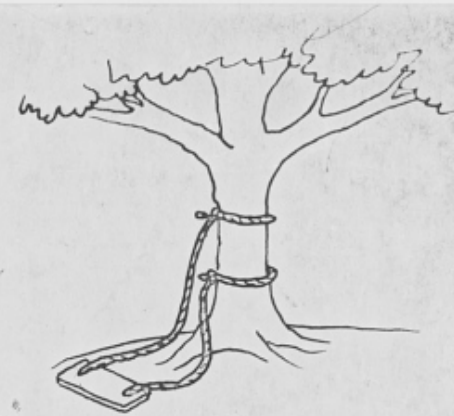
- Thayer:
  - Cori pg. 171-182 “Fundamentals of Master Scheduling”,
  - Fairley 183-194 “Work Breakdown Structures”



AS MARKETING REQUESTED IT



AS SALES ORDERED IT



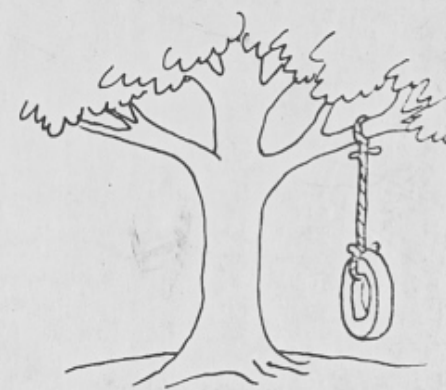
AS ENGINEERING DESIGNED IT



AS WE MANUFACTURED IT



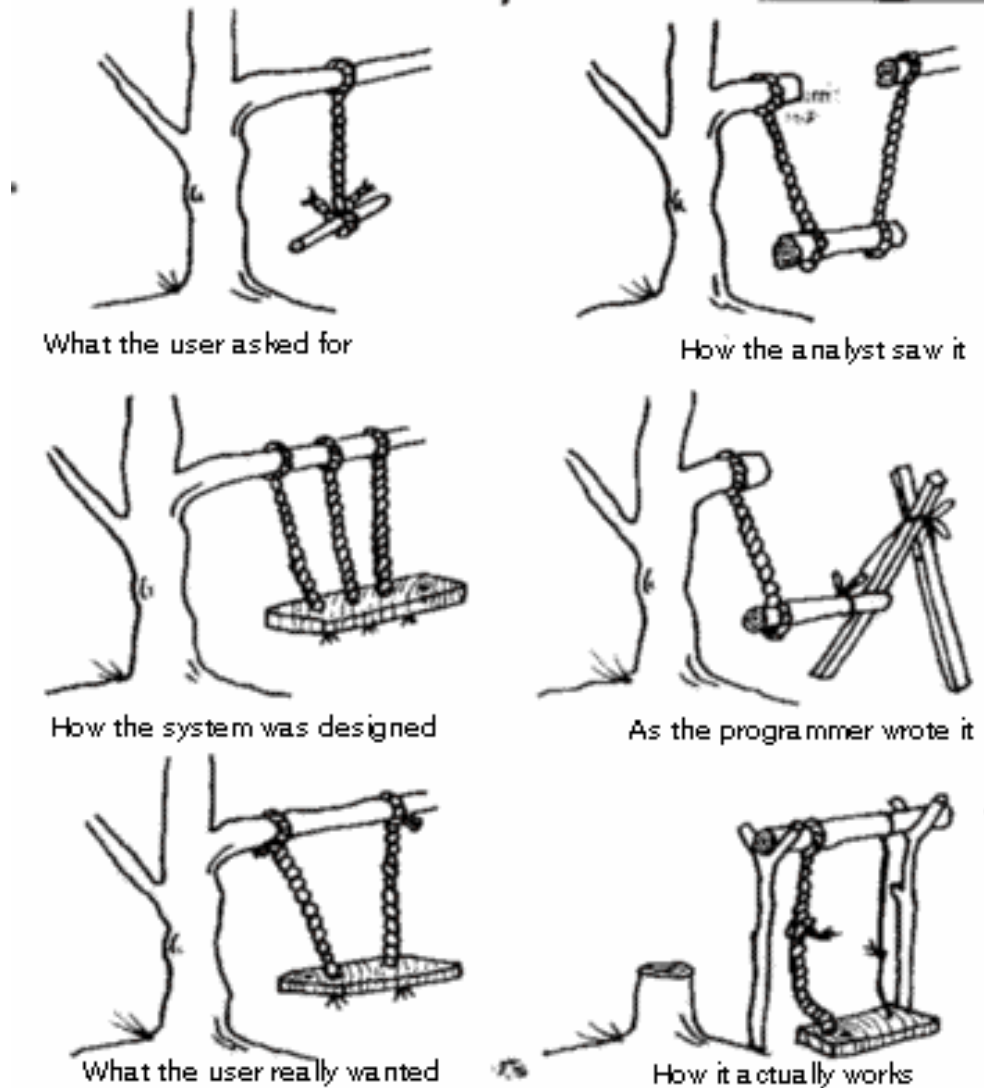
AS FIELD SERVICE INSTALLED IT



WHAT THE CUSTOMER WANTED!!!

"COMMUNICATION" MEANS: SAYING AND HEARING HAVE THE SAME MESSAGE

**Tree Swing picture from 1970s - Businessballs.com (Ack T & W Fleet)**



[source <http://www.cs.ucl.ac.uk/external/atanu/req.gif>]



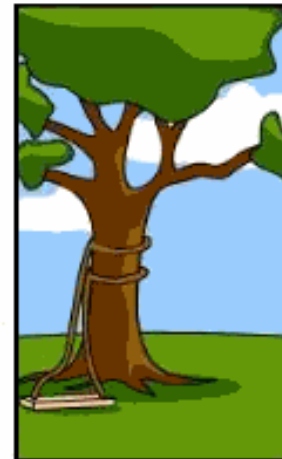
How the customer explained it



How the Project Leader understood it



How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



How the project was documented



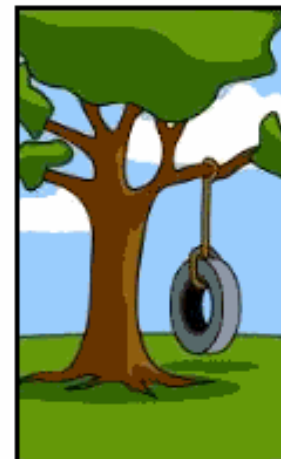
What operations installed



How the customer was billed



How it was supported



What the customer really needed

[source <http://www.codinghorror.com/blog/images/software-engineering-explained.png> ]

- <http://www.businessballs.com/treeswing.htm>