

MacroEconometric Forecasting



Topic:

Distinguishing experiments from other empirical methods

Presented by Munisa Yashnarbekova

SDLC: Systems Development Life Cycle



- SDLC is a *Life Cycle*
- All systems have a life cycle or a series of stages they naturally undergo.
 - The number and name of the stages varies, but the primary stages are conception, *development*, maturity and decline.
 - The systems development life cycle (SDLC) therefore, refers to the development stage of the system's life cycle.
- A Life Cycle

Methodologies



- Is there a difference between the term SDLC and the term 'methodology'?
- SDLC: refers to a stage all systems *naturally* undergo,
- A Methodology refers to an approach *invented by humans* to manage the events naturally occurring in the SDLC.



Methodologies

- A methodology is, a set of steps, guidelines, activities and/or principles to follow in a particular situation.
- Most methodologies are comprehensive, multi-step approaches to systems development
- There are many (thousands!) methodologies available. See www.methodology.org .



SDLC vs. Methodology

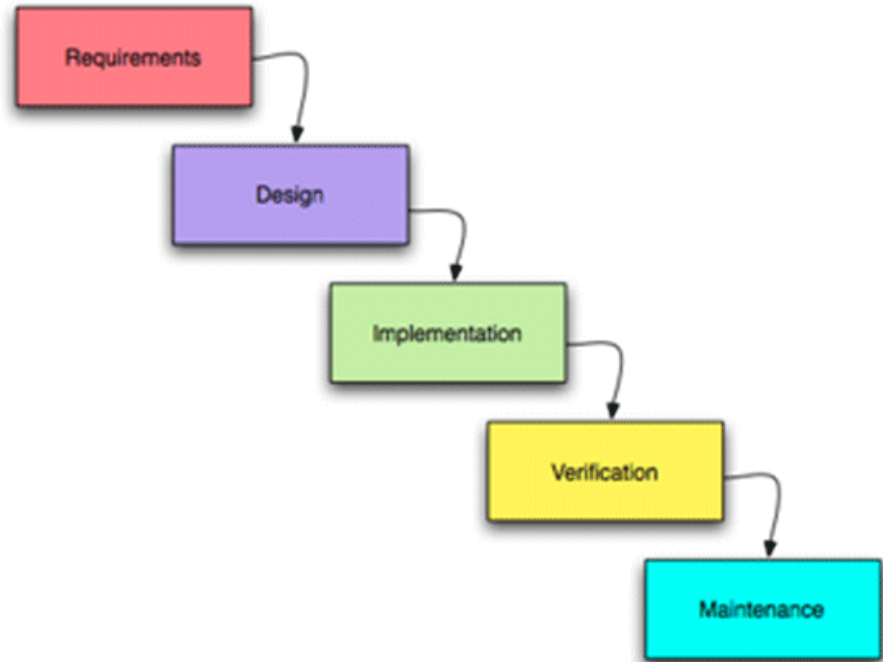
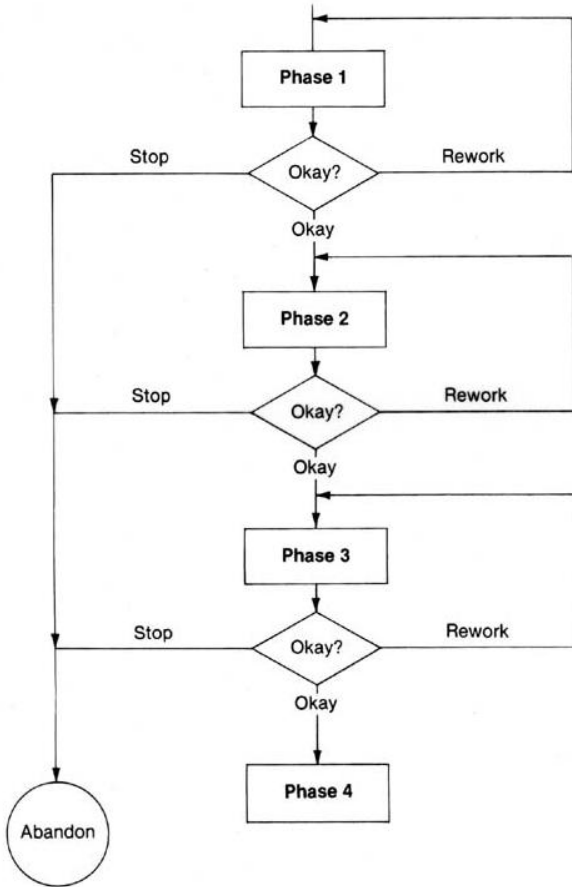
- The term SDLC is frequently used synonymously
- with the waterfall or traditional approach for
- developing information systems.

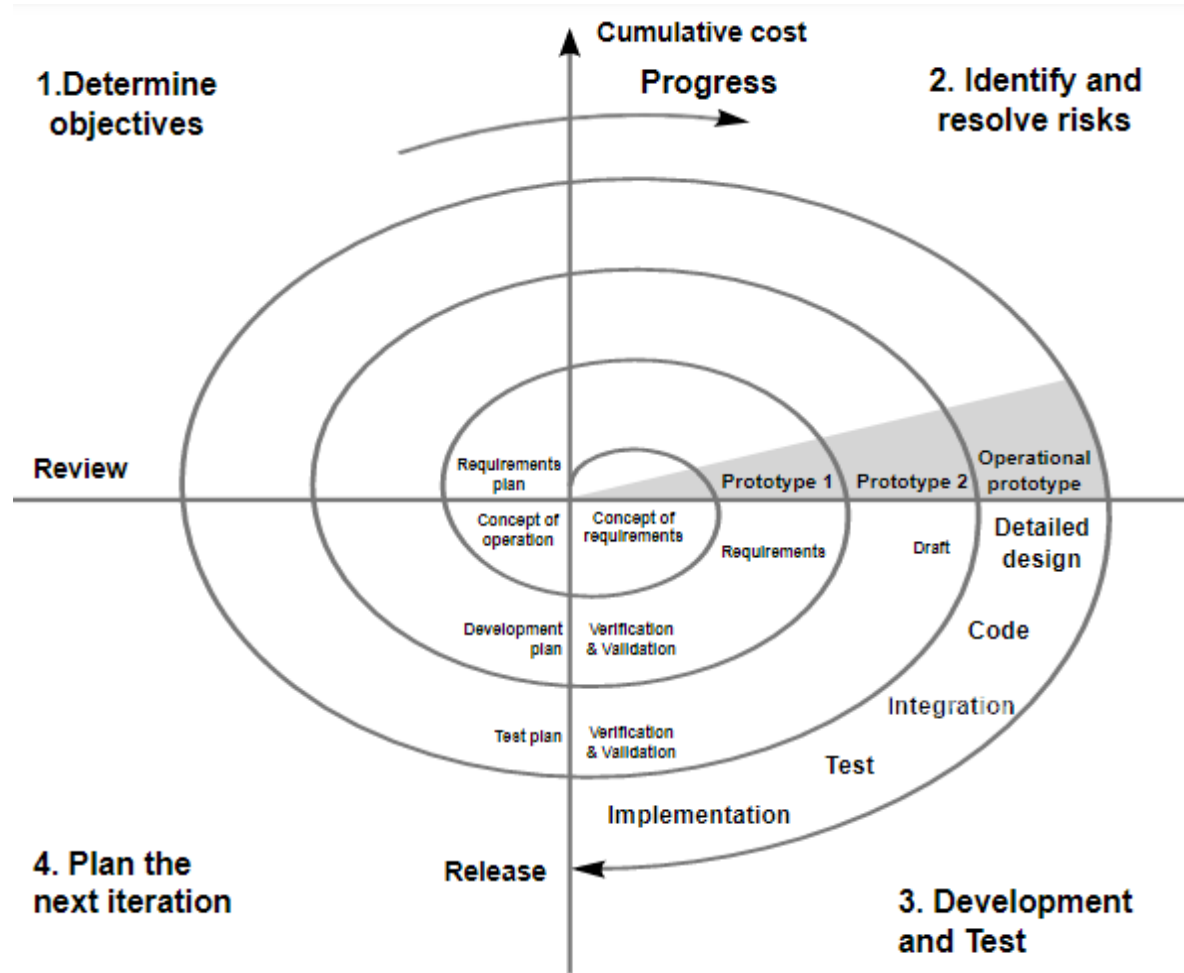
• **WRONG !!!**

“The Waterfall approach”



- This approach essentially refers to a linear sequence of stages to develop a system from planning to analysis to design to implementation.
- Stages are followed from beginning to end.
- Revisiting prior stages is not permitted.







Spiral development

- Combination of waterfall methodology and prototyping methodologies
- Combination of the top-down *and* bottom up concepts

- Primary focus is on risk assessment
 - Break the project into smaller segments
 - Controls the impact of change



Spiral model, continued

- Each cycle includes the same stages
 - As identified in SDLC model

- Each cycle includes
 - Determine objectives, alternatives and constraints
 - Evaluate alternatives and resolve risks
 - Develop and verify deliverables
 - Plan the next iteration

-

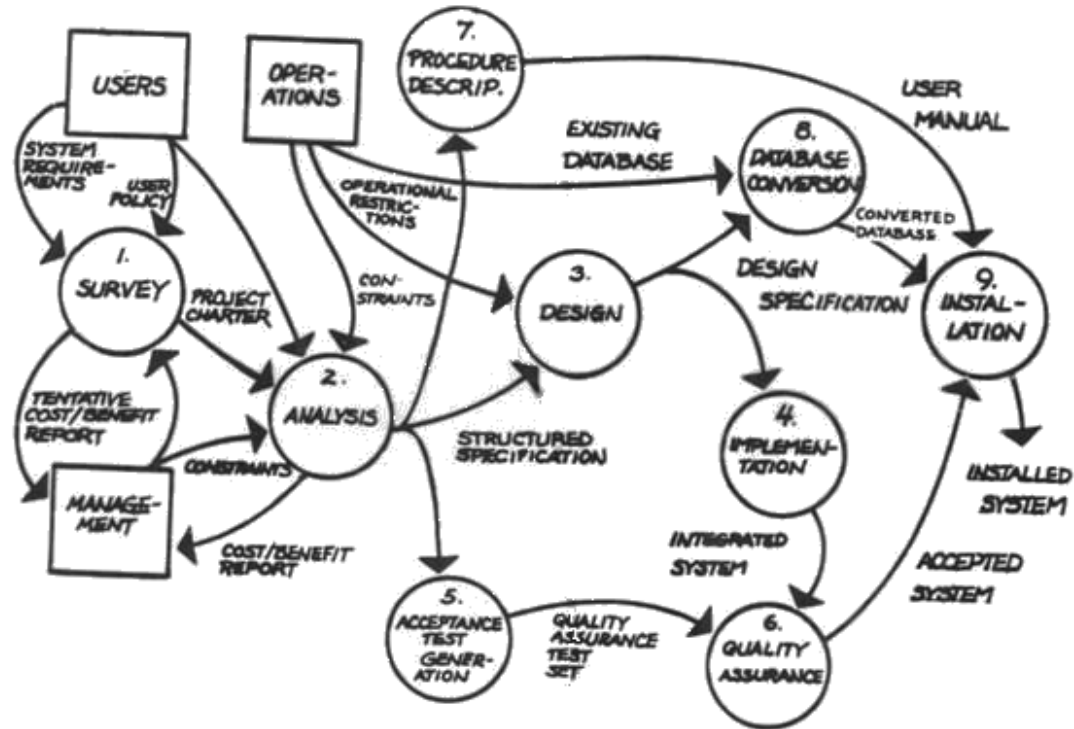
Approaches to Systems Development



- **Process-Oriented Approach**
 - Focus is on flow, use and transformation of data in an information system
 - Involves creating graphical representations such as data flow diagrams and charts
 - Data are tracked from sources, through intermediate steps and to final destinations
 - Natural structure of data is not specified



A Process Oriented Methodology





Process oriented approach

- Look at all of our processes
- Processes take precedence over data
- Get the processes correct first
- Worry about the data after

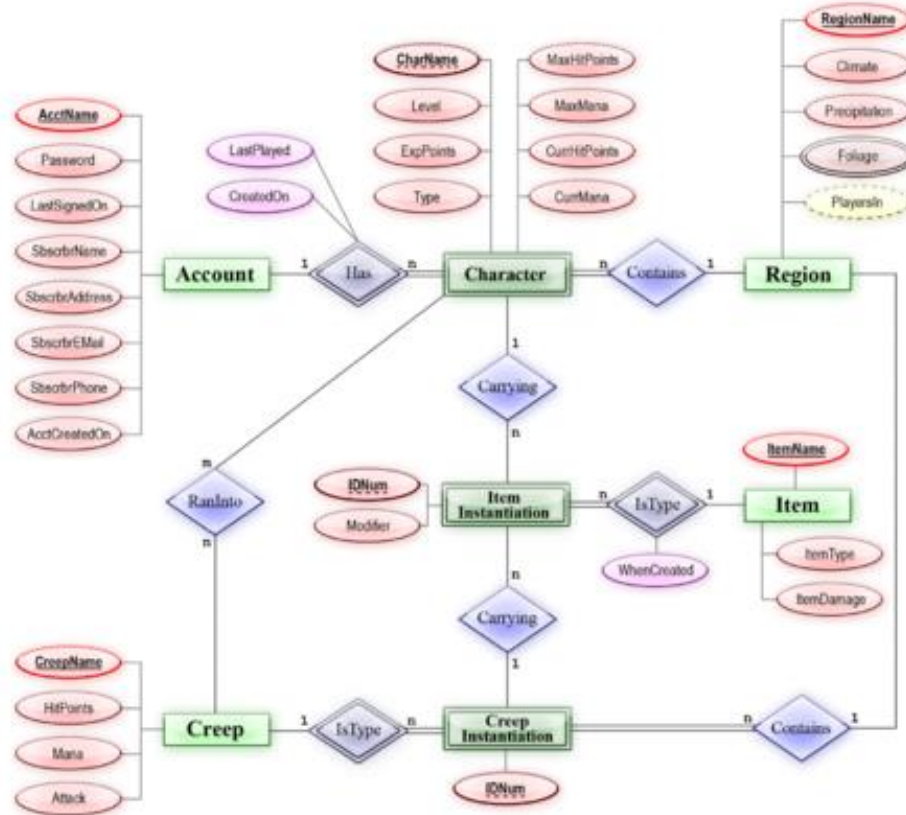


A Data-oriented approach

- Depicts ideal organization of data, independent of where and how data are used
- Data model describes kinds of data and business relationships among the data
- Business rules depict how organization captures and processes the data
- Create User Stories



Data Oriented methodology





Data oriented approach

- Forget the processes, look at the data
- Data comes first, get the data correct
- Once the data have been defined correctly, look at how the processes actually use the data

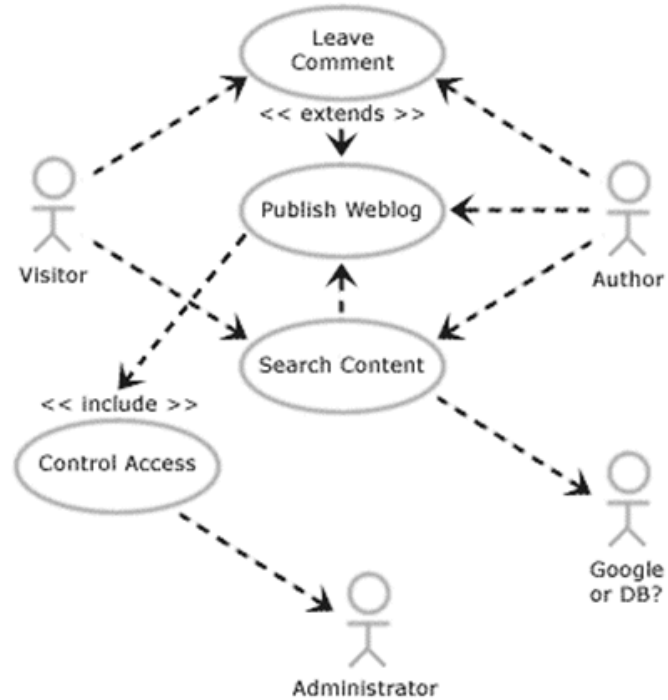


Object Oriented Approach

- Define the objects that are part of the system first
- Identify use cases; What happens and what are the responses to those actions
- Build the remainder of systems once use cases are identified



Object Oriented Analysis





Systems Development Life Cycle

- Every methodology has roughly the same stages
- Usually they stages are
 - Project Identification and Selection
 - Project Initiation and Planning (just after Conception)
 - Analysis
 - Design
 - Implementation
 - Maintenance (starting Maturity)



Systems Development Life Cycle

- Maintenance
 - System changed to reflect changing conditions
 - System obsolescence

A good way to learn the stages of the SDLC is to create deliverables (output) of each stage in the process.



Additional methodology types

- Prototyping
 - Building a scaled-down working version of the system
 - Advantages:
 - Users are involved in design
 - Captures requirements in concrete form
- Rapid Application Development (RAD)
 - Utilizes prototyping to delay producing system design until after user requirements are clear

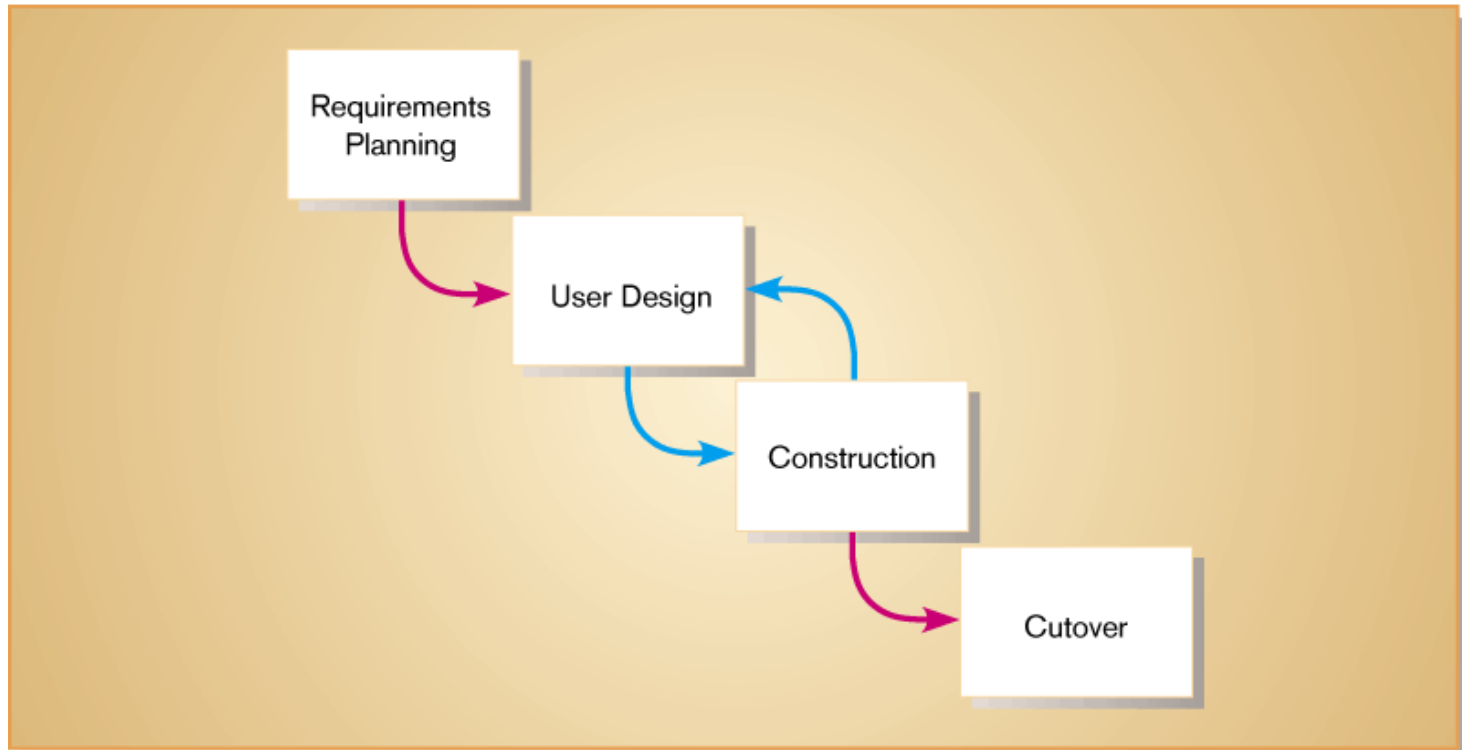
Joint Application Design (JAD)



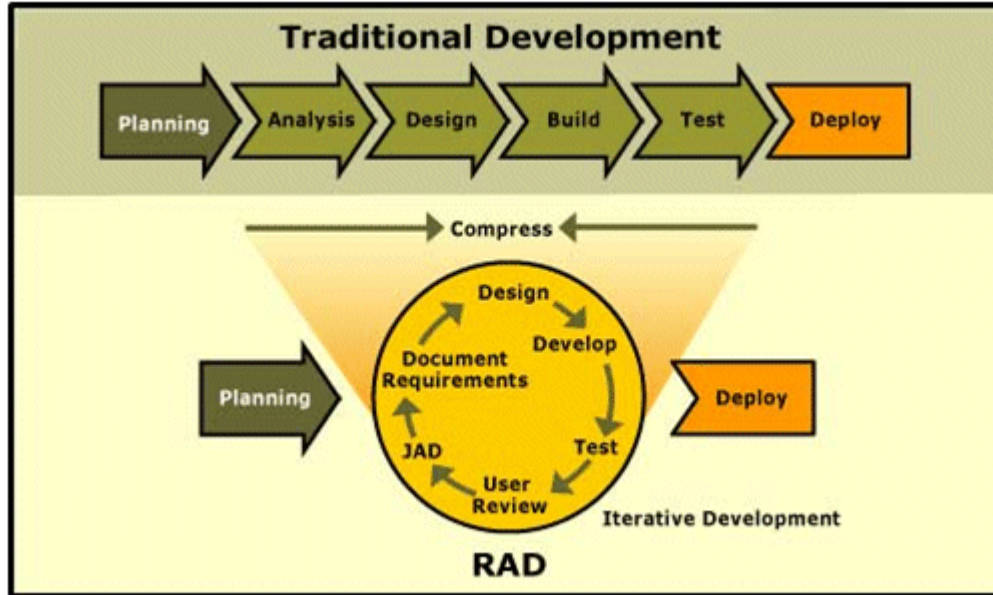
- Users, Managers and Analysts work together for several days
- System requirements are reviewed
- Structured meetings



Prototyping



RAPID APPLICATIONS DEVELOPMENT





AGILE METHODS

What is XP?

- XP is a lightweight methodology for small to medium sized teams developing software in the face of vague or rapidly changing requirements.

-- Kent Beck.

- XP is:

- Humane.
- Honest.
- Productive.
- Professional.
- Fun.



Developing at the Speed of Change





XP

- XP is a discipline of software development.
- There are certain things you must do.
 - You must write tests before code.
 - You must program in pairs.
 - You must integrate frequently.
 - You must be rested.
 - You must communicate with the customer daily.
 - You must follow the customer's priorities.
 - You must leave the software clean and simple by the end of the day.
 - You must adapt the process and practices to your environment.



Why can XP be better?

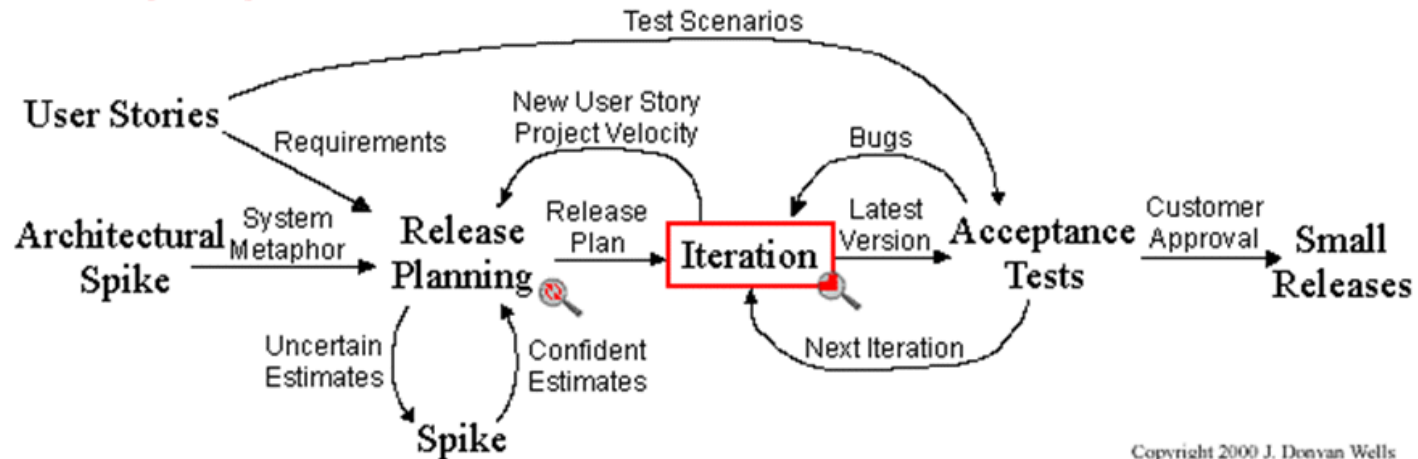
- Its early, concrete, and continuing feedback from short cycles.
- Its incremental planning approach, which quickly comes up with an overall plan that is expected to evolve through the life of the project.
- Its ability to flexibly schedule the implementation of functionality, responding to changing business needs.
- Its reliance on automated tests to demonstrate the presence of features.
- Its reliance on oral communications, tests, and source code to communicate system structure and intent.



Extreme Programming Project



Extreme Programming Project

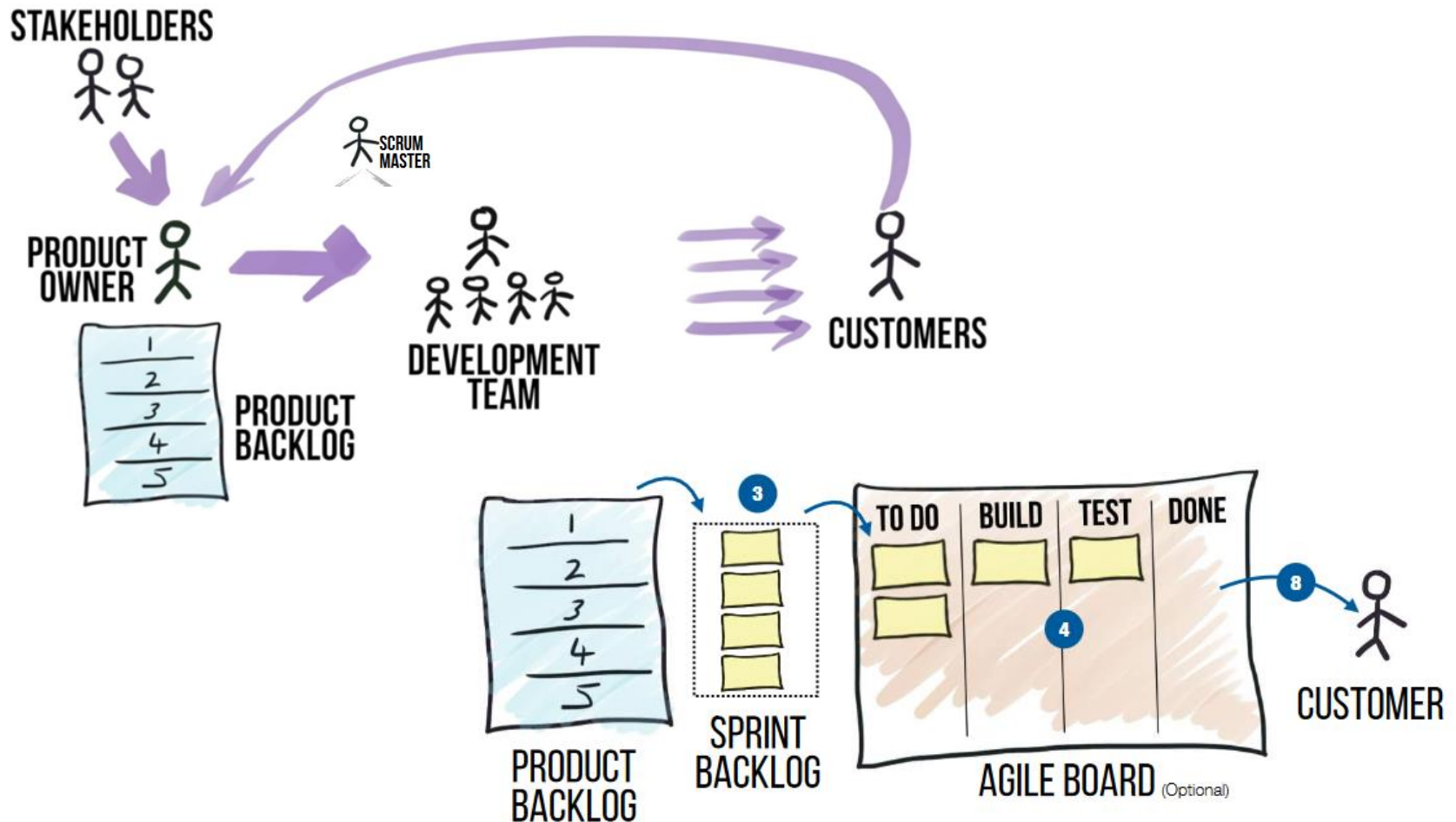


Copyright 2000 J. Donovan Wells

For more specific definitions of agile methodologies check out [Agile's Interview questions and Answers](#).



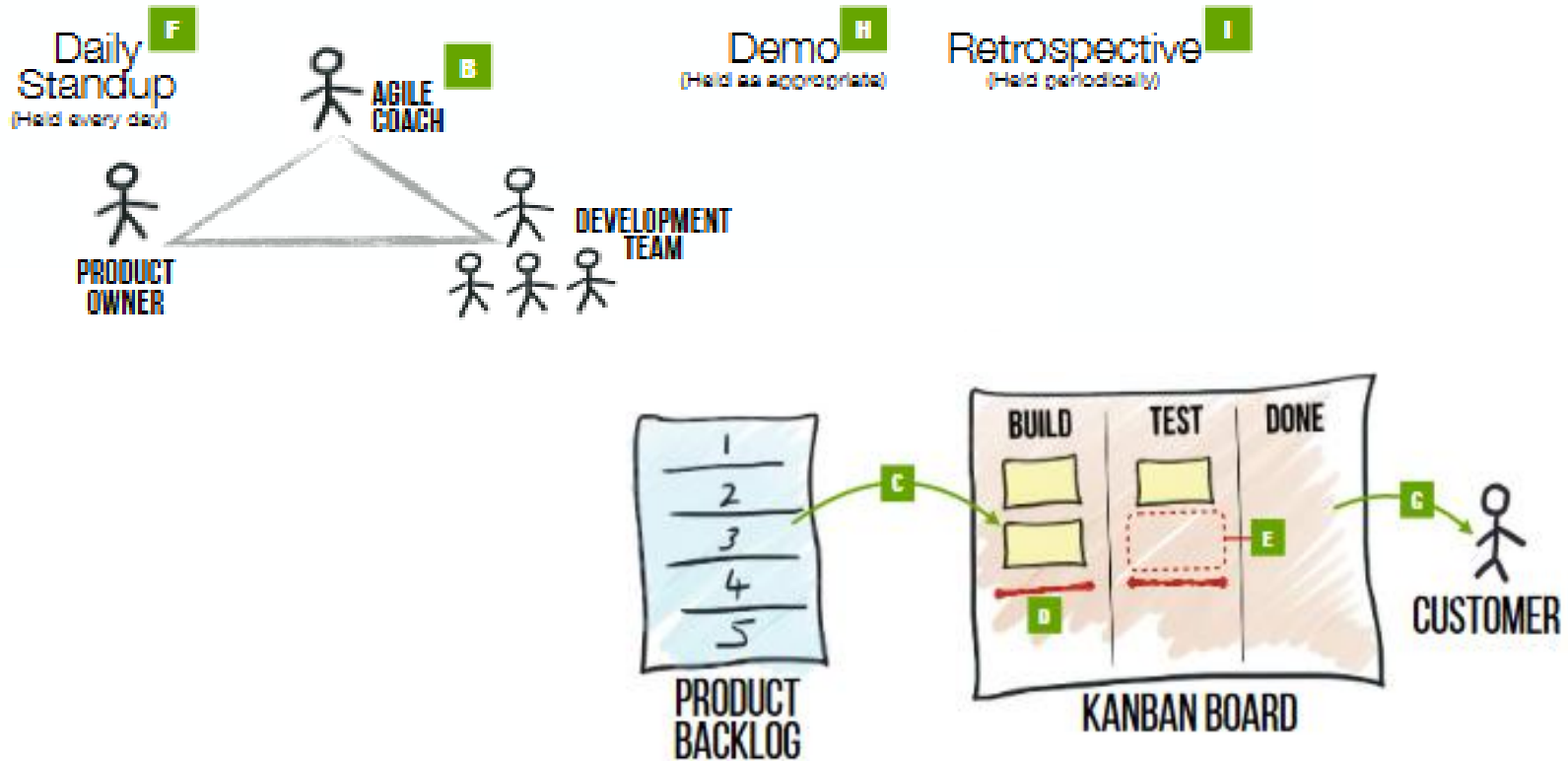
Scrum (agile) Methodology



Material was borrowed from the “development that pays” site. Material is available at [that site](#) and is used with permission.

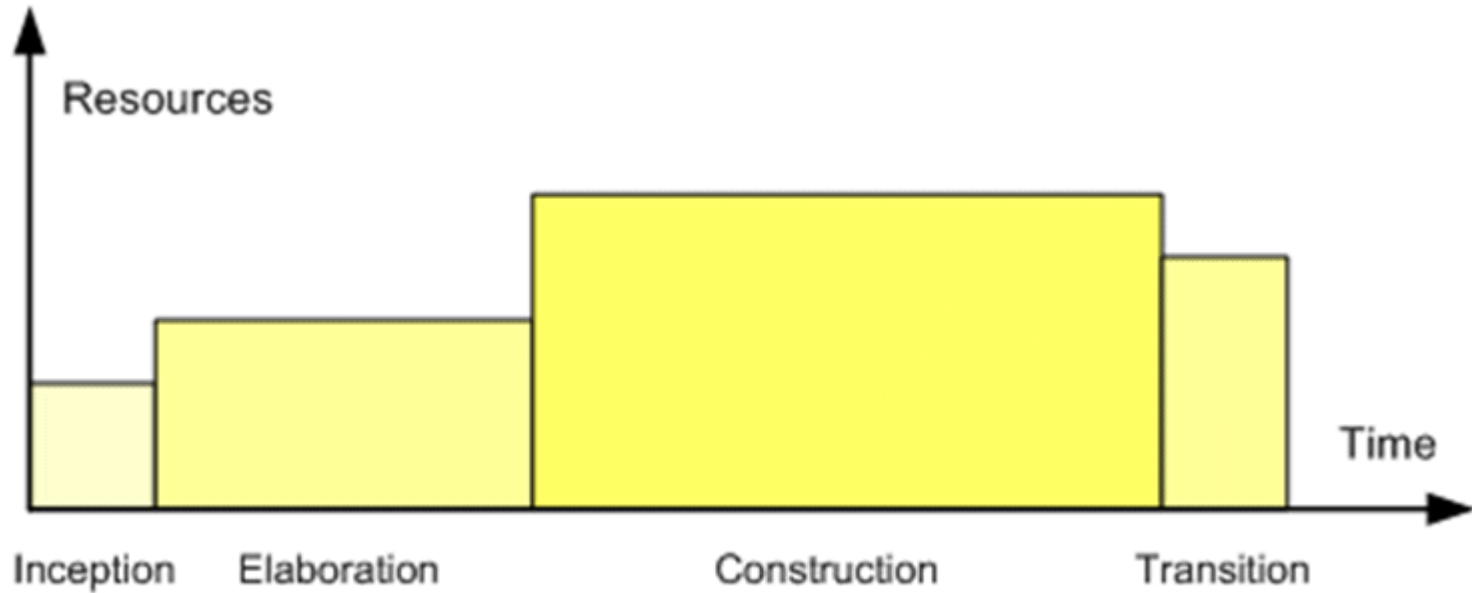


Kanban (agile) Methodology



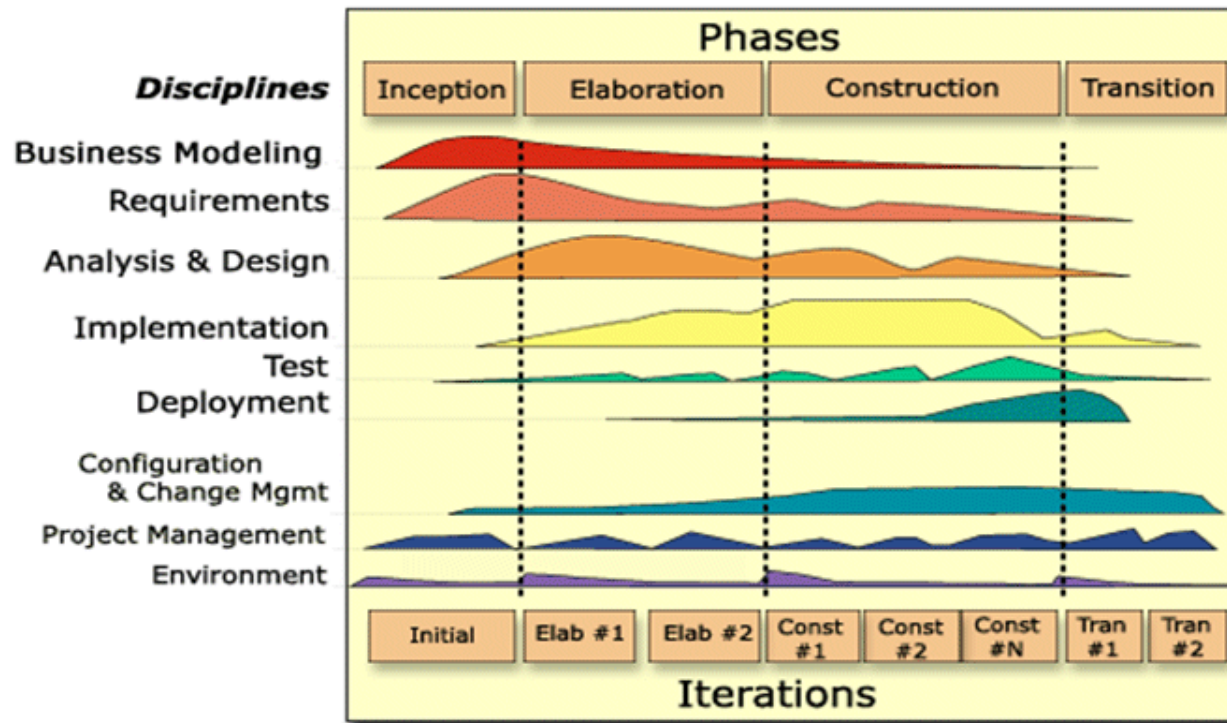
Material was borrowed from the “development that pays” site. Material is available at [that site](#), and is used with permission.

Rational Unified process





Rational unified process



Joint application design



- Users, Managers and Analysts work together for several days
- System requirements are reviewed
- Structured meetings

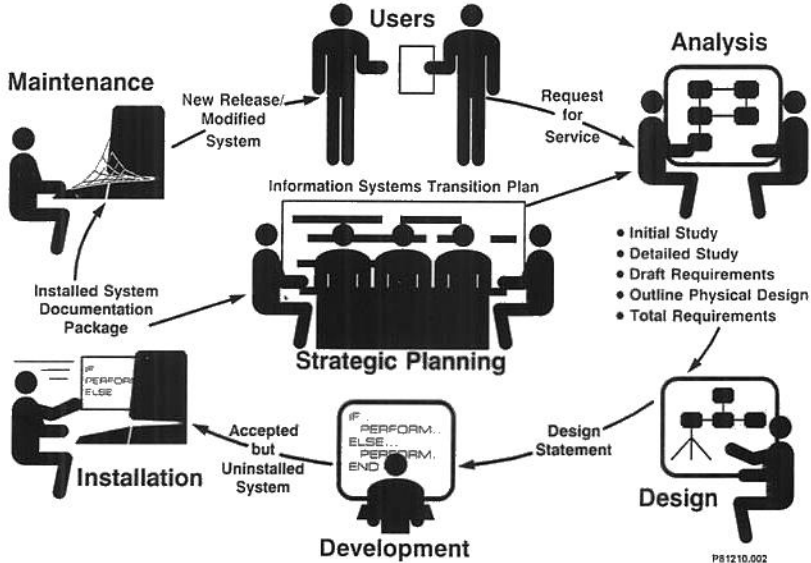


Risks of Agile and Prototyping

- Premature convergence
- The design is not technically feasible
- The design phase encouraged a “big batch”
- The prototype “tests well” but fails to truly meet user needs
- You miss opportunities to deliver value (and learn) earlier
- You miss opportunities to engage engineers early in ideation
- The end product does not fully leverage what is technically achievable

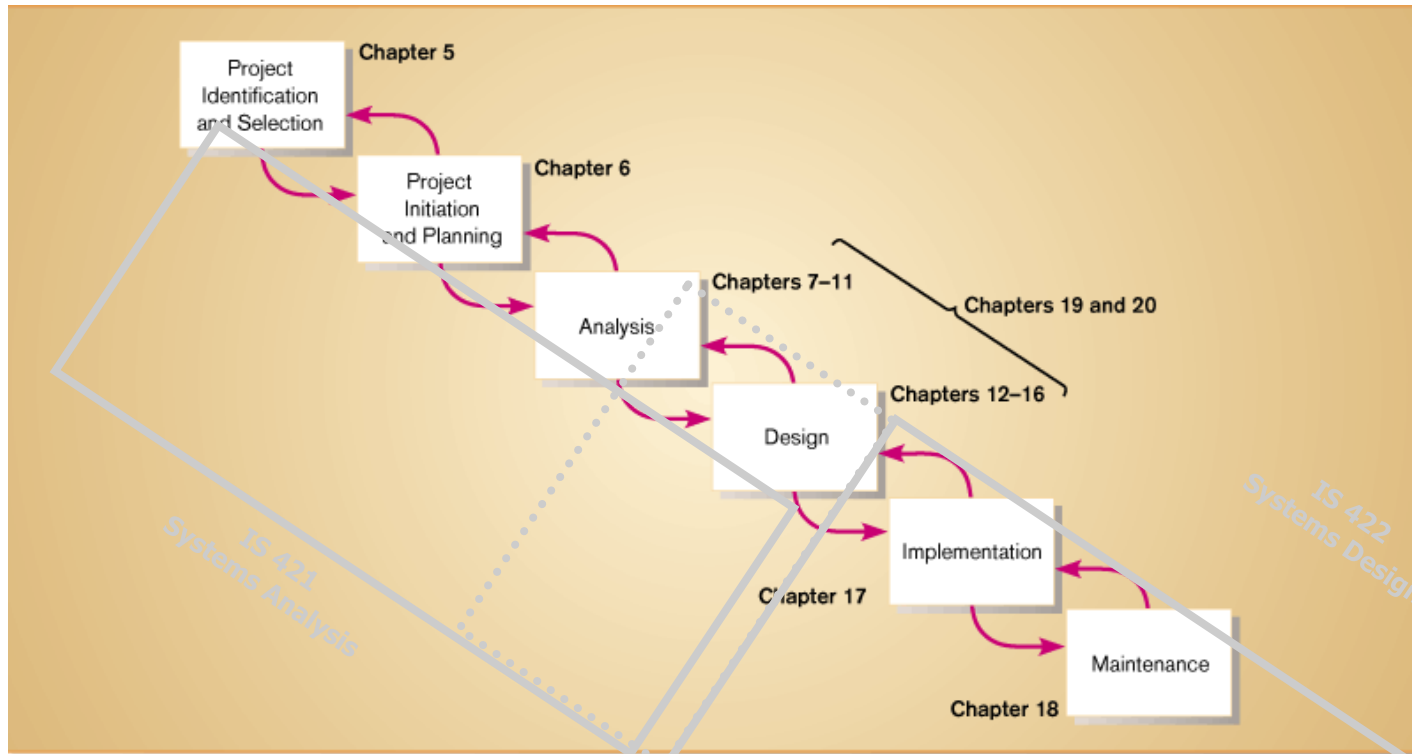
From John Cutler, *Is Agile the Enemy (of Good Design)?*
Hackernoon, [Available Online](#). Last viewed July 31, 2018.

Enterprise system planning





Stages of the SDLC



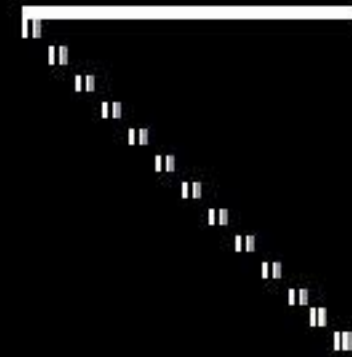
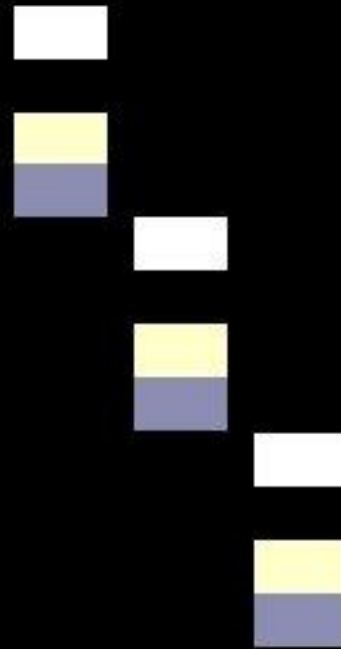
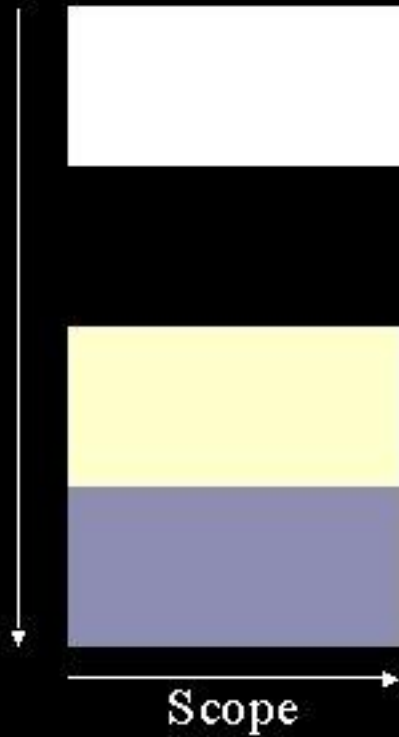


Waterfall

Iterative

XP

Time



Scope

Phases of the Systems Development Life Cycle



- **Project Identification and Selection** Two Main Activities
 - Identification of need
 - Prioritization and translation of need into a development schedule
 - Helps organization to determine whether or not resources should be dedicated to a project.
- **Project Initiation and Planning**
 - Formal preliminary investigation of the problem at hand
 - Presentation of reasons why system should or should not be developed by the organization

Systems Development Life Cycle



- Analysis
 - Study of current procedures and information systems
 - Determine requirements
 - Study current system
 - Structure requirements and eliminate redundancies
 - Generate alternative designs
 - Compare alternatives
 - Recommend best alternative



Systems Development Life Cycle

- Design
 - Logical Design
 - Concentrates on business aspects of the system
 - Physical Design
 - Technical specifications
- Implementation
 - Implementation
 - Hardware and software installation
 - Programming
 - User Training
 - Documentation



- Discussion about each methodology and its benefits is available at
 - <http://www.technologyuk.net/software-development/sad/methodologies.shtml>
- This resource has recommendations about the various stages.



Roles of a Methodology

- Specify *what tasks* need to be completed
- Specify *what deliverables* should be created
- Specify *whom* should be included



How the customer explained it



How the Project Leader understood it



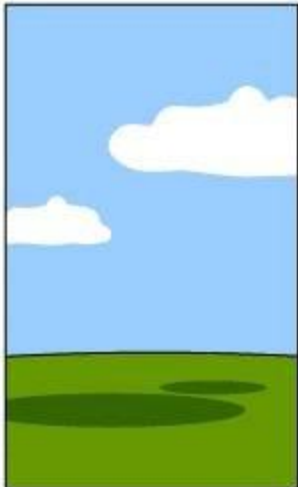
How the Analyst designed it



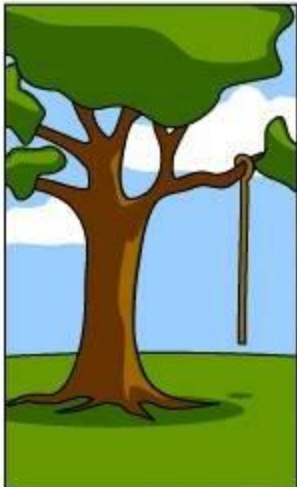
How the Programmer wrote it



How the Business Consultant described it



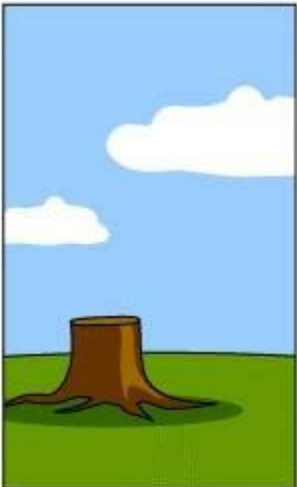
How the project was documented



What operations installed



How the customer was billed



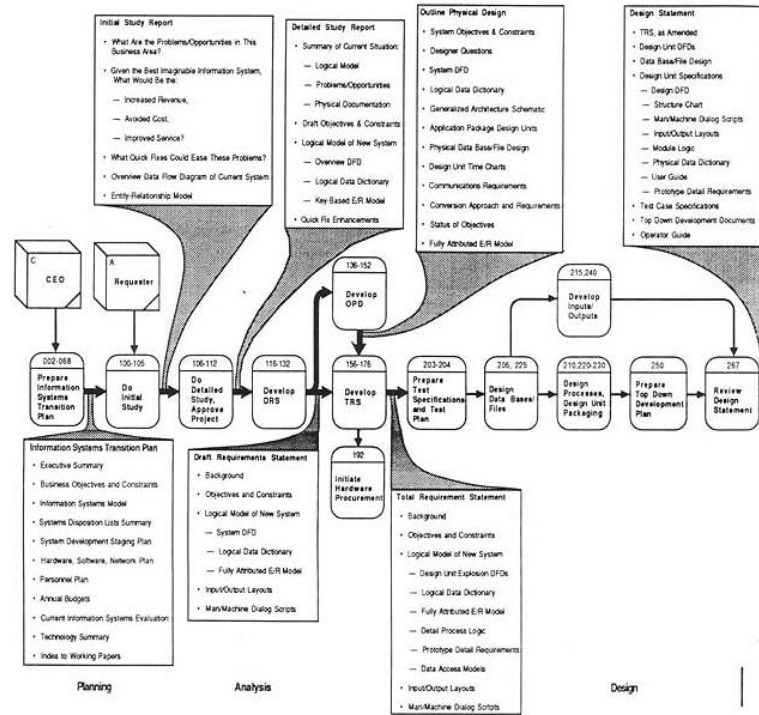
How it was supported

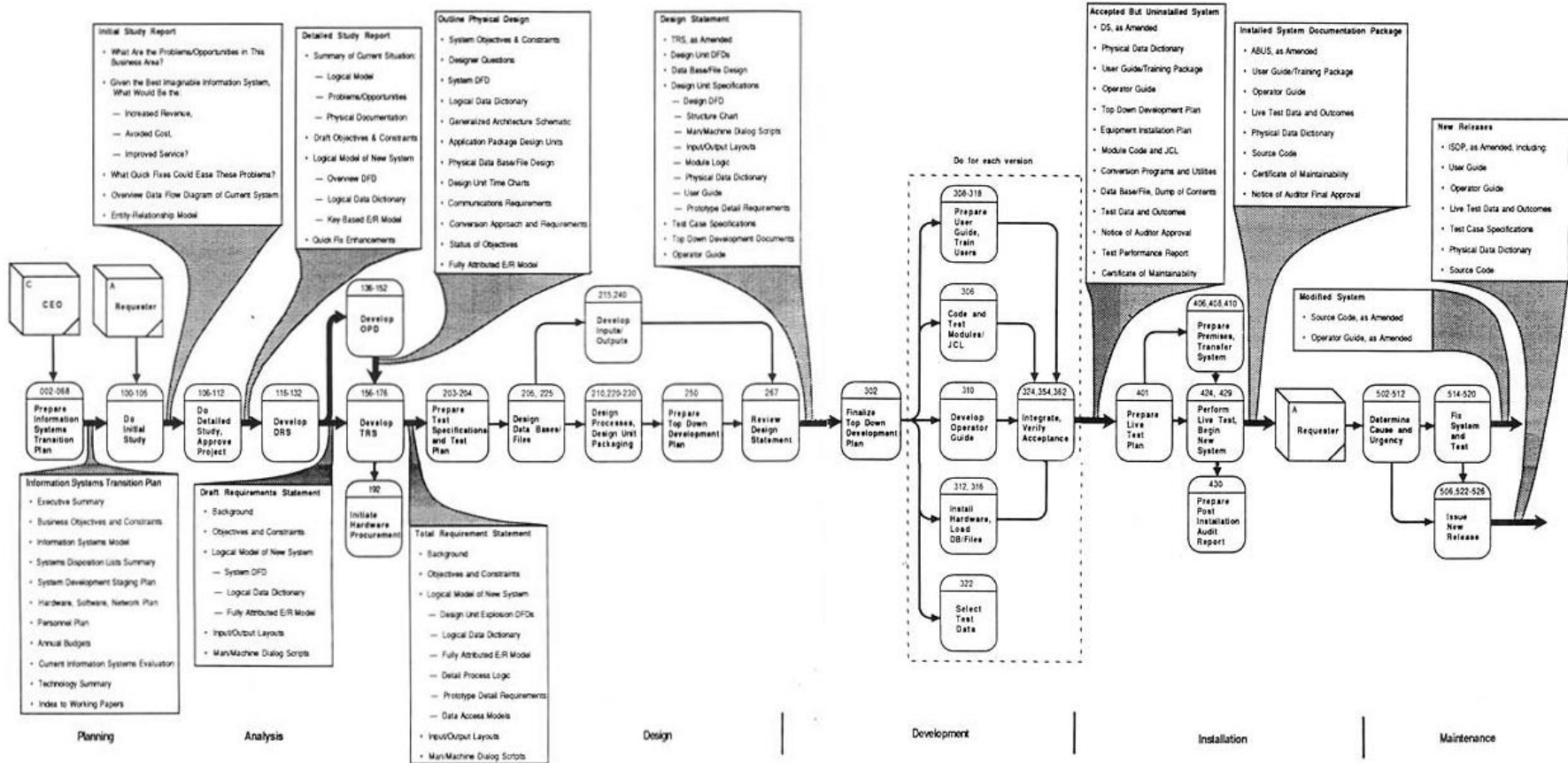


What the customer really needed



What a real methodology looks like







Role of a Methodology

To identify and track necessary deliverables

DELIVERABLE PLANNING GUIDE			Project Types					
STRADIS Workproducts with Table of Contents Id	Final /Activity	ENH	New Applications					
			SML	TYP	LRG	4GL	PKG	CSH
Initial Study Report								
1	Summary	/103		Do	Do	Do	Do	Do
2	Terms of Reference Memo	/100	Do	Do	Do	Do	Do	Do
3	Details of Interviews	/102	Inf	Inf	Do	Do	Do	Do
4.1	Problems	/102	Inf	Inf	Do	Do	Do	Do
4.2	Opportunities	/102	Inf	Inf	Do	Do	Do	Do
5	Overview DFD (current)	/102			Do	Do	Opt	Opt
6	Entity-Relationship Model	/102	Inf	Inf	Do	Do	Do	Do
	List of DSR Vendors	/105			Opt	Opt		
	Maint/Enhancement Request	/105			Opt	Do	Opt	Opt
	SteerBdy Invitation Letter	/105		Inf	Do	Do	Do	Do
	Work Authorization for DSR	/105	Inf	Inf	Inf	Do	Inf	Inf

Legend: ENH = Enhancement/Maintenance, SML = Small, TYP = Typical, LRG = Large, 4GL = Fourth Generation language, PKG = Purchased Software, CSH = Crash
 Do = Workproduct, Inf = Informal, Opt = Optional, " " = Don't do



Role of a methodology

- Project Management Guidelines

Phases	Deliverables (What)	Activities/Roles (When/Who)	Techniques (How)	
Planning Analysis Design Development Installation Maintenance	Major - Minor - Minor - Minor	Activities Activities • • • Activities	Roles Responsibility Entering Leaving Tasks Define Redefine Finalize	Accelerated Analysis Prototyping DFDs E/R Modeling Reverse Engineering 4GL Normalization etc.
Quality				
Project Management				



Role of a methodology

- Estimation Guidelines

Estimating Schemes That Don't Work

The ideal estimating tool would enable the planner to answer a number of questions about the problem to be solved, and then based on historical data about previous projects, compute the cost and duration of the project. No such tool exists! Many schemes of this nature have been proposed over the last 20 years, and sold to gullible managers; none has proved of value.

Several schemes exist for determining programming cost based on number of lines of code to be programmed. These are of no value because:

- the number of lines of code is not a good measure of the difficulty of a problem, and programmers can proliferate lines of code at will.
- the number of lines-of-code cannot be estimated with much reliability until the programming has been done, by which time it's too late to be of use as an estimating tool.

There is, as yet, no substitute for the step-by-step estimation of tasks, based on Raw Estimate Worksheets. McDonnell Douglas has and will continue to, evaluate a variety of estimating schemes; if and when any estimating tool of value is developed, it will be built into STRADIS. (See the Estimating Guidelines section in the Techniques I volume.)



Role of a methodology

Estimating - Rules Of Thumb

Some rules of thumb have evolved; these are to be taken as purely indicative, and should always be revised for each organization in the light of its specific experience.

Management

a) Project management overhead: half-a-day per week per person directly supervised.

Analysis

b) Interviewing: 12 hours per interview. Allow half-a-day to plan and carry out an interview, half-a-day to analyze results, half-a-day to document and review results.

c) Data Flow Diagrams:
- Analysis and draft DFD 1 hour per 1 process per version.
- Final DFD .1 hour per 1 process

d) Data element specification and normalization: 1-2 hours per element

e) A page of structured English: 4 hours per page
A page of tight English: 1 additional hour

f) Writing a page of narrative report from known facts: 1-2 hours per page

g) Finalizing a report layout given good user cooperation: 4-8 hours per report

h) Specifying a test case from business logic: 1-2 hours per case
Writing test data from specification and projecting results: 2 hours per case



Role of a methodology

Design

- i) Developing the Outline
Physical Design: 20-40 man-days
- j) Design DFD:
- Analysis and draft DFD .5 hour per 1 process per version
- Final DFD .1 hour per 1 process
- k) Structure Charts: .1 hour per 1 module for each version
- l) Laying out a screen: 4-8 hours per screen

Programming

- m) Simple report programs
assuming time-sharing
facility and existing data
definitions to include coding,
testing and debugging: 30-hours COBOL
4-8 hours Report Writer
- n) Given an extremely tight pro-
gram specification develop
program to include coding,
testing and debugging in a
high-level language: 4 hours per structure chart module
- o) Conducting a structured
walkthrough: 2-5 hours per participant (1 hour to
prepare, 1-4 hours for walkthrough)



Role of a methodology

Recognition Of Indicators Of Potential Failure

The Executive Sponsor, Project Director, Technical Project Manager, and Steering Body members should be on the alert to indicators that 'signal danger'. Indicators are not always read correctly; there may be a real problem. Following are only some of the most common indicators, not all of them.

Unrealistic Schedule	Has time been provided for vacations, sick leave, and other contingencies? Is work week assumed to be 100% productive? Does anybody have too many overlapping activities?
Overextended project manager	Is he/she trying to do too many things; still performs some old line management duties, working long and extra hours, assigned too many tasks to him/herself, minimized time required to manage project, not provided time for product reviews and personnel counseling?
Lack of Acceptance	Has the Project Director or Technical Project Manager elected to continue without the appropriate approvals or without conditions attached?
Slipped Dates	Does it seem like missed target dates is becoming the way of life? An occasional slip is expected, but not many. Is the statement made: "no problem, we can make it up"?
Lack of User Involvement	Has the user become uninterested? Has he not fulfilled his commitment on resources?
Non-use of Change Control Procedures	Have you discovered changes to the original scope that were accepted by project people without proper approval?
Unauthorized Plan Deviation	Have changes been made to the plan without proper authorization, no matter how small?
Defeatist Attitude	Are the team members making statements that indicate they feel their efforts are in vain, that the project is headed for failure?
Breakdown of Team Spirit	Are team members starting to take sides in disputes, using terms - we and they, or he and I - and writing memos that appear to accomplish nothing but cover their hindside? Does morale appear to be low?

Skill /Role Matrix	Skill /Role Matrix																																		
	Application Analyst	Auditing Consultant	Chief Financial Officer	Chief Information Officer	Communications Director	Communications Manager	Data Administrator	Data Base Designer	Direct User	Executive Sponsor	Hardware Operator	Hardware Planning Specialist	Human Resource Manager	Human Subsystem Implementer	IT Planning Team Leader	Maintenance Manager	MG Training Coordinator	Performance Acceptance Specialist	Planning Project Specialist	Planning Project Director	Programmer	Programming Team Consultant	Project Director	Requester	Reviewing Auditor	Staffing Body	Systems Designer	Systems Expert	Systems Manager	Technical Consultant	Technical Project Manager	Test Analyst	Test Coordinator	User Representative	
Drawing OFDs	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Building Data Dictionary	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Data Analysis /Modeling	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Specifying Logic	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Identifying Objectives	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Develop Business Functional Model	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Drawing Design OFDs	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Strategic Design	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Develop Information Systems Model	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Drawing Structure Charts	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Top Down Development Planning	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Structured Walkthroughs	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Information Gathering	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Data Base /File Design	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Develop H /3 /N Plan	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Develop Personnel Plan	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Develop Staging Plan	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Man /Machine Dialog	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Programming	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
User Guide Design	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Security and Control	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Prototyping	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Package Selection	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Project Planning	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Project Tracking	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Project Reporting	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Change Control	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Cost /Benefit Analysis	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Progress Reporting Individual	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Develop Future Sys Architecture	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Develop Annual Budget	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Specifying Test Cases	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Testing	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

Legend
 F = Frequency
 P = Priority



Role of a methodology

- Consistency of product
- Ease of reusability of code
- Facilitates changes in personnel
- Consistency of documentation

Reference and source



- Conceptual Econometrics Using R (ISSN Book 41) 1st Edition, by Hrishikesh D. Vinod (Editor)
- Principles of Macroeconometric Modeling (Volume 36) (Advanced Textbooks in Economics, Volume 36) by L.R. Klein, W. Welfe, et al. | Oct 5, 1999
- Macroeconomic Modeling and Macroeconometric Simulation: Illustrated with a developing economy Model (Macroeconometric model Book 1) Book 1 of 1: Macroeconometric model | by Kannapiran Arjunan | Jun 9, 2020
- Global and National Macroeconometric Modelling: A Long-Run Structural Approach by Anthony Garratt, Kevin Lee, et al. | May 4, 2012
- Simulation of a macroeconometric model with multiple time series considerations (Wayne economic papers) by Rosemary Rossiter | Jan 1, 1982