- Common vocab
  - Project manager
    - Analyst with diverse skill set responsible for project initiating planning executing closing down
  - Project
    - Planned undertaking of related activities to reach an objective that has a beginning and an end
  - Deliverable
    - End product of an SDLC phase
  - SDLC(system development life cycle)
    - Traditional methodology used to develop maintain and replace info systems
  - Feasibility study
    - A study that determines if proposed info system makes sense for org economically
  - Project work book
    - An online or hard copy repository for all project correspondance
  - Gantt chart
    - Graphical representation of a project that shows each task as a horizontal bar
- Data flow diagram- DFD
  - Key concept
    - Balancing
      - Inputs and outputs that appear on level 0 must be preserved during decomposition
    - Decomposition
      - Iterative process of breaking down the description of a system into finer and finer detail
      - Primitive DFD
  - Four symbols
    - Data flow
      - Data In motion, moving from one place in a system to another
    - Data store
      - Data at rest, represent physical location for data
    - Processes
      - Work or action preformed on data so that they are transformed, stored, or distributed
      - 4 main processes
        - Caputering data from different sources
        - Maintening data stores
        - Producing and distibuting data to different sinks
        - High level descriotions of data transformation operations
      - Can be coupled or decoupled
        - Coupled
          - Rely on each other for information
        - Decoupled
          - Independent of each other
    - Sources/sink
      - Origin or destination of the data
  - Level - 0 diagram
    - Represents the primary individual processes at the highest possible level
  - Level - n diagram
    - Represent a DFD generated from a n nested decomposition from a level 0 diagram
  - Rules
    - Input to a process are different from the outputs
    - Objects have unique names
    - Processes
      - Cannot have only outputs
      - Cannot have only inputs
      - Has a verb phrase label
    - Data stores
      - Data cannot move directly from one data store to another

  - Data cannot move directly from an outside source to an data store
  - Data cannot move directly to an outside sink from a data store
  - Has a noun phrase label
- Source/sink
  - Data cannot move directly from a source to a sink
  - Has a noun phrase label
- Data flow
  - Has only one direction between symbols
  - A fork in data flow means that data goes from a common location to separate location
  - A join in data flow means that data goes from separate locations to a common location
  - Cannot go directly back into the same process it just left
  - Data flow to a data store means update, delete or change
  - Data flow from a data store means retrieve or use
  - Has to have a noun phrase label

- Advanced rules
  - A composite data flow can be split into component data flows at the next level but no new data
  - The inputs to process must be sufficient to produce the output from process
  - Lowest level DFD new data may be presented to represent data transmitted under exceptional conditions
  - To stop data flow tangling you may repeat data stores or souces on a DFD

- Use cases
  - Show the behavior or functionality of a system
    - Consist of a set of possible sequences of interactions between a system and user
  - Symbols
    - Actor
      - User role, represented by stick figures, outside system boundary
    - Use case
      - Single system function represented by ellipse
    - System boundary
      - Represented by a box that includes all relevant use cases
    - Connections
      - Actors connected with solid lines
      - Use cases connected with arrows
      - <<extend>>
        - Arrow points use case extended
        - Association between use cases where one adds new behaviors and actions
      - <<include>>
        - Arrow points use case included
        - Association between use cases where one uses the functionality of the other
      - Instantiated
        - Concrete not abstract, with values and parameters
  - Level
    - Level of detail
      - High general abstract
      - Low detailed
      - 5 levels
        - White
          - As seen from clouds
        - Kite
          - More detailed but still flying
        - Blue
          - Sea level

- Fish
  - Below sea level with lots of detail
- Black
  - Maximum detail provided

  - Trash
    - Stake holders
      - People who have vested interest in the system being developed
    - Preconditions
      - Things that must be true before use case can start
    - Minimal guarantee
      - Least amount promised to the stakeholder by a use case
    - Success gurantee
      - What a use case must do effectively to satisfy stakeholder
    - Trigger
      - Event that initiates a use case
- Sequence diagram
  - Depicts interactions among objects during a certain period of time
  - Two forms
    - Generic
      - All possible sequences of interactions
    - Instance
      - Sequence for one scenario
      - UML
      - Path
        - Specific combination of conditions within the use case
        - Vertical axis
          - Represents time
          - Time increases as we go down vertical
        - Horizontal axis
          - Represents various participating objects
          - Shown as a vertical dash line called life line
          - Object symbol
            - Box with objects name underlines
  - Activation
    - Time period during which an object performs an operation
    - Represented by a superimposed rectangular box on the life line
  - Communication
    - Objects communicate by sending messages
      - Shown as a solid arrow from sending object to receiving object
  - Synchronous message
    - Show as a full solid arrowhead
    - A type of message in which the caller has to wait for the receiving object to finnish executing before It can finnish
  - Simple message
    - Shown by a transverse tick mark
    - A message that transfers control from the sender to recipient
  - Asynchronous message
    - Shown as a half arrowhead
    - A message in which the sender does not have to wait for the recipient to handle the message
  - Guard condition
    - Ensures the message will only be sent if condition met
  - X
    - Object dead

- Relational database systems model
  - Data represented as a set of related tables or relations
  - Relation
    - A named two dimensional table of data
      - Each relation consists of a named columns and arbitrary number of unnamed rows
      - Each column corresponds to an attribute of that relation
      - Each row corresponds to a record that contains data values for an entity
  - Primary key of the relation
    - Identifier attribute
  - Well structured relation
    - A relation that contains minimum amount of redundancy and that allows user to insert modify and delete rows without error inconsistencies
  - Normalization
    - Process of converting complex data structures into simple stable data structures
  - Advanced rules
    - Second normal form (2nf)
      - Each non primary key attribute is identified by the whole key full functional dependency
    - Third normal form (3nf)
      - Non primary key attributes do not depend on each other no transitive dependency
  - Functional dependency
    - A constraint between two attributes in which the value of one attribute is determined by the value of another
  - Foreign key
    - Attribute that appears as a non primary key attribute in one relation and a primary key in another relation
  - Description
    - Entry cells are simple, intersection of row and column have a single value
    - Entries in a given column are from the same set of values
    - Each row is unique
    - Sequence of columns is interchangeable
    - Rows are interchangeable
- Gantt chart
  - Graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion
  - Do not show how task must be ordered
  - Show when task should begin and when they should end
  - Time overlap

- Network diagram
  - Is a critical path method
  - Diagram that depicts project task and their interrelationships
  - Showa ordering of task
  - No time flow
  - Shows which could be done in parallel no over lap

- PERT
  - Program evaluation review technique
    - Uses optimistic, pessimistic and realistic time estimates to calculate the expected time of task
    - Optimistic (o),
      - Minimum time
    - pessimistic (p)
      - Maximum time
    - realistic (r)
      - PM best guess

- ET
  - $(O+4r+P)/6$

CPM -crtitical path method

- construct a model of the project that includes the following:
  - A list of all activities required to complete the project
  - The time (duration) that each activity will take to complete,
  - The dependencies between the activities and,
  - Logical end points such as milestones or deliverable items.
- CPM calculates the longest path of planned activities to logical end points or to the end of the project, and the earliest and latest that each activity can start and finish without making the project longer.

Cocomo

- Constructive cost model
- Estimates projects size and cost
  - Uses parameters derived from prior models of different complexity
  - predict human resources requirement for basic, intermediate, and very complex system
- Appropriation of human skill to task at hand
  - Language, speed ect

Project size / duration

- Number of individual projects and length of time needed to complete

Value chain analysis

- Analyzing an organization activities to determine where value is added to products and or service and the cost incurred for doing so; usually also include a comparison with the activities added value and costs of other organizations for the purpose of making improvements in the organizations operation and performance
- Extent to which activities add value and cost when developing products and service
- Inputs to organization
- Output products service

Break even analysis

- At what point if ever do benefits = cost
- PV
  - Present value
  - PV n=Ydollars (1/(1+i)nthpower
- NPV
  - Net present value
  - Subtract one time cost and present value recurring cost from present value yearly benefits
- Break even
  - (Yearly NPV cash flow --- overall NPV cash flow)/ yearly NPV cash flow

PADIM

- Planning
- Analysis
- Design
- Implementation
- Maintenance

Traditional waterfall

- Flows downhill
  - Planning -> analysis -> logical design -> physical design -> implementation -> maintenance
- Little feed back between phases

CASE

- Computer aided
- Code generators enable automatic generation of program and database definition code

RAD

- Rapid application development
- Fast, user involvement, prototyping, integrated case tools and code generators
- Requirements planning -> user design <-> construction -> cutover (delivery)

Agile

- Focus on adaptive rather than predictive methodologies
- Focus on people rather than roles
- Focus on self adaptive processes
- Recomended for unpredictable/dynamic requi responsible motivated developers customers involved

Extreme programming

- Short cycles incremental planing approach focus on automated tests reliances on evolutionary approach to development that lasts throughout the life time of system
- Two person programing team on site customer
- Planning analysis design and construction are fused into one phase -> unique way of capturing and presenting system requirements and design specifications
- Code while testing, coding testing listening and designing
- Better communication among developers , higher level productivity , higher quality code, test discipline

RUP

- Rational unified process - interactive, incremental approach to systems development
- OOAD
- Inception-define scope, feasibility requirement plan ->elaboration- detail user requirments base line architecture ->construction- coded tested documented ->transition- deployed users are trained and supported

IETC

- a. Divide things into basic units then fuse back together

NTG (nominal group technique)

- Individuals working together to solve a problem in name only
- Work alone, written list of ideas ->end of idea generation time, members pool individual ideas ->open discussion -> idea reduction

JAD (joint application design)

- Structured process where users managers analyst work together for several days in a series of intensive meetings to specify or review system requirements
- JAD session leader -> users ->manager ->sponsor ->system analyst ->scribe ->IS staff

Prototyping

- Iterative process of system development in which requirements are converted to a working system that is continually revised through close collaboration between an analyst and user
- Diagram initially put together ->system construct ->function+ data ->activate ->modify as needed <->reactivate as needed
- Evolutionary prototyping
  - Begin by modeling parts of target system, if successful evolve rest of system from parts
  - Refined until completion
- Throwaway prototyping
  - Does not preserve the prototype
  - Demonstrate some aspect of the system that is unclear, helps users decide different features or characteristics
  - Discarded

Software

- Outsourcing
  - Turning over responsibility for some or all of an organizations info system application and operations to an outside firm
- IT services
  - Custom systems
- Packaged software producers (off shelf)

- Pre made software with limited modifiability
- ERP enterprise resource planning
    - System that integrates individual traditional business functions into series of modules so that a single transaction occurs seamlessly within a single info system rather than several separate systems
- Cloud computing
    - Provisions of computing resources, apps over the internet, no investment needed  to run and maintain the resources
- Open-source
    - Free software created by caring communities
- In house
    - Made by your organization

ISP -information system planning

- Orderly means of assessing the information needs of an organization and defining the information systems databases and technologies that will best satisfy those needs
- Three steps
    - Step 1
        - Current situation
            - Listing of manual and automated processes
            - Listing of manual and automated data
            - Technology inventory
            - Human resources inventory
    - Step 2
        - future situation
            - Blueprints of manual and automated processes
            - Blueprints of manual and automated data
            - Technology blueprints
            - Human resources blue prints
    - Step 3
        - Schedule of project
            - Mapped and graphed schedules of projects
                - Gant charts and shit
- top down planning
    - High level organizational perspective with active involvement of top level management
- Bottom up planning
    - Faster less costly but may fail to view informational needs of entire organization
    - Requires identification of business problems and opportunities that are used to define projects

## Wireless distributed systems

How it works

- access point also known as a bridge has the capability to relay traffic from one segment to another. It performs this task with the use of a “Bridge Learn Table”, where MAC addresses are stored in association with the LAN segment (or physical interface) where they reside (from the perspective of the bridge).
- Traffic between wireless LAN devices that conform to the IEEE 802.11 standard require 4 MAC addresses instead of 2.
    - the MAC address of the sender,
    - the MAC address of the final destination,
    - the MAC address of the sending PC card in the access point,
    - MAC address of the receiving PC card in the other access point.

- Roaming between cells that are interconnected by a WDS link works exactly the same as for cells that are interconnected via Ethernet. The effect of a relocation of a station from one cell to the other is that the Bridge Learn Tables will be updated to reflect the new location of the station. This is done by the hand-over request messages that are part of the IAPP (Inter Access Point Protocol).

Pros

- Cost effective:
  - No additional expense in terms of adding a wireless link to an already installed Access Point. Adding a WDS link merely requires a reconfiguration of the Access Point, without having to pay the price for an additional PC Card
- Flexible:
  - Expanding an existing wired infrastructure network by adding coverage for office space that is not adjacent to the existing office can be easily achieved, providing great flexibility.
    - WDS is also an excellent solution to create a roaming network in an area where wired connections between the APs cannot be established.

Cons

- Encryption:
  - It is not possible to use encryption with dynamic assigned and rotating keys, on the WDS link. Only fixed assigned WEP keys can be used to provide encryption.
- Performance:
  - Use of a single PC Card (and a single channel) results in sharing the same channel amongst the Access Points and the clients, the end-to end throughput will be less than the maximum attainable value. Obviously using a second PC card can improve this situation but in that case the expense of a second card has to be accepted.
- Outdoor operation:
  - WDS allows creation of point to point connections, which would suggest that this could be applied to outdoor installations as well. Though in principle this is true, one has to remember that the IEEE802.11 standard has been devised primarily for LAN (indoor) operations, and that for use in outdoor situations (especially long distances and point to multi-point configurations) additional provisions are to be implemented.

Possible configuration

- StarConfiguration:
  - WDS links are established between one AP and several others. The central AP could be part of a wired infrastructure network, while the “satellite” APs are positioned to cover an area which is larger than can be covered by a single cell, on all sides of the central AP
- ChainConfiguration:
  - Where the Star configuration can cover a more rectangular or square area, a Chain configuration allows coverage of a longer shape (for instance a long corridor). The AP’s are chained together, where the first AP for example could have a connection to the existing infrastructure (with all the network resources).

Issues

- If a single PC card is used for all wireless traffic and wireless clients are operational as well, end-to-end throughput may be considered too low (depending on the applications).
- In the chain configuration, if the chain becomes very long end-to-end latency issue might come into play.

virtual private network (VPN)

- extends a private network across a public network, such as the Internet. It enables a computer to send and receive data across shared or public networks as if it were directly connected to the private network, while benefiting from the functionality, security and management policies of the private network
- done by establishing a virtual point-to-point connection through the use of dedicated connections, encryption, or a combination of the two.

- To prevent disclosure of private information, VPNs typically allow only authenticated remote access and make use of encryption techniques.
- VPNs provide security by the use of tunneling protocols and through security procedures[7] such as encryption. The VPN security model provides:
  - confidentiality
    - such that even if the network traffic is sniffed at the packet level (see network sniffer and Deep packet inspection), an attacker would only see encrypted data
  - sender authentication
    - to prevent unauthorized users from accessing the VPN.
  - message integrity to
    - detect any instances of tampering with transmitted messages
- Authentication
  - Tunnel endpoints must authenticate before secure VPN tunnels can be established.
  - User-created remote-access VPNs may use passwords, biometrics, two-factor authentication or other cryptographic methods.
  - Network-to-network tunnels often use passwords or digital certificates. They permanently store the key to allow the tunnel to establish automatically, without intervention from the user.

## OSI protocols -- Open Systems Interconnection protocols

- family of information exchange standards
- 7 layers
  - Layer 1: physical layer
    - This layer deals with the physical plugs and sockets and electrical specification of signals only.
    - This is the medium over which the digital signals are transmitted. It can be twisted pair, coaxial cable, optical fiber, wireless, or other transmission media.
  - Layer 2: data link layer
    - The data link layer packages raw bits from the physical layer into frames
    - This layer is responsible for transferring frames from one host to another. It might perform error checking. This layer further consists of two sublayers : MAC and LLC
  - Layer 3: network layer
    - This level is in charge of transferring data between systems in a network, using network-layer addresses of machines to keep track of destinations and sources. This layer uses routers and switches to manage its traffic (control flow control, error check, routing etc.) So here it takes all routing decisions, it deals with end to end data transmission.
    - Examples
      - Connectionless Network Service (CLNS)
      - Connectionless Network Protocol (CLNP)
      - Connection-Oriented Network Service (CONS)
      - Connection-Oriented Network Protocol
      - Network Fast Byte Protocol
  - Layer 4: transport layer
    - The transport layer transfers data between source and destination processes.
    - two connection modes are recognized, connection-oriented or connectionless.
    - Connection-oriented service
      - establishes a dedicated virtual circuit and offers various grades of guaranteed delivery, ensuring that data received is identical to data transmitted.
    - Connectionless mode
      - provides only best-effort service without the built-in ability to correct errors, which includes complete loss of data without notifying the data source of the failure.

Layer 5: session layer

The session layer controls the dialogues (connections) between computers.

It establishes, manages and terminates the connections between the local and remote application.

It provides for full-duplex, and half-duplex or simplex operation, and establishes checkpointing, adjournment, termination, and restart procedures.

Layer 6: presentation layer

This layer defines and encrypts/decrypts data types from the application layer.

Protocols such as MIDI, MPEG, and GIF are presentation layer formats shared by different applications.

Layer 7: application layer

The application that requesting connectivity and using it such as internet explorer or mail services

## DISTRIBUTED GRAPHICS SYSTEMS

No clue but heres random information

Pattern Recognition

classification, which attempts to assign each input value to one of a given set of classes such as determining whether a given email is "spam" or "non-spam". Pattern recognition algorithms generally aim to provide a reasonable solution for all possible GUI inputs, and a "fuzzy" matching of inputs is done. This is compared using pattern matching algorithms, which look for the matches of the inputs with the pre-existing patterns. The results of pattern recognition comparison is very accurate due to the set of algorithms that matches the scanned inputs to the existing set of characteristics to determine what type of object it is

The Net-Manager Concept

Existing programs should need only small adaptations for working inside the net, the net specific parts should be encapsulated from the rest of the program.

Programs adapted for the net-manager should still be able to run stand-alone without the net-manager in the background.

Programs should be able to exchange arbitrary data, not only some predefined data types the net-manager knows.

It should be possible to run programs stand-alone first and then start the net-manager or connect to an already running net-manager.

The concept should be extendable to hierarchies of nets with local net-managers on different machines.

The net-manager should be easily configurable for the user.

More net manager stufff

The net-manager is the control instance for building and maintaining the net of programs. Programs can be added to and removed from the net by starting them or telling them to quit, the machines they run on and the displays they use can be configured. Information about the available programs, machines and displays is provided by the user.

Programs running under the control of the net manager communicate with the net-manager over a permanent command connection which is created when the program is started by the net manager or when a running program connects itself to the net for the first time and deleted when the program is disconnected from the net.

To send or receive information programs can create input and output ports which are registered with the net-manager. Each port has a fixed type which determines what kind of data it is able to handle.

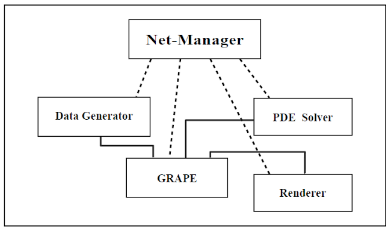
The net-manager coordinates and supervises all data exchange in the net. The net-manager creates information channels between programs by connecting their input and output ports. The port types are used to determine which connections are valid. The programs are informed that their ports are connected, but the knowledge about the net of connections is kept within the net-manager.

## MAPREDUCE Operation

- Cluster mapping of initial structured data sets in parallel nodes using distributed algorithm
- Filtering of primary data into a secondary repository
- Controlled by master servers at each step
- Applications include
  - sequential graphics data reduction implementing filtering mechanisms thereby creating and rendering a high resolution composite graphic on a much smaller data set
  - Routing financial data for transactional processing using optimal filtering

## CLOUD SYSTEMS

- Top reasons for cloud computing adoption
  - 66% faster deployment of infastructure
  - 68% hardware systems
  - 57% reduce systems management burden
- Five characteristics of the cloud
  - On demand and self service
  - Broad network access
  - Resource pooling
  - Rapid elasticity
  - Measured service
- Cloud computing service models
  - User cloud (SaaS) (SOFTWARE AS A SERVICE)
    - Single application, multi-tenancy, network based, one to many delivery of application all users have access to same features
    - Example --Salesforce.com, Google Docs, Red Hat Network/RHEL



  - Development cloud (Paas)(PLATFORM-AS-A-SERVICE)
    - Application developer model, Application deployed to an elastic service that autoscales, low administrative overhead. No concept of virtual machines or operating system.
    - Code it and deploy it.
    - Example-- Google AppEngine, Windows Azure, Rackspace Site, Red Hat Makara
  - Systems cloud (IaaS) (INFRASTRUCTURE-AS-A-SERVICE)
    - Servers and storage are made available in a scalable way over a network.
    - Example-- EC2,Rackspace CloudFiles, OpenStack, CloudStack, Eucalyptus
  - Deployment models
    - Public
      - Amazon web services
    - Private
      - In house cloud
    - Hybrid
      - Vpn?
  - Cloud requires Architectural design
    - Cloud Computing isn't a “magical solution”
    - Need to design your architecture with the end in mind
    - As you build it make your infrastructure easily replicable
  - Automation unlocks the potential of the cloud
    - Devops and agile IT philosophy
    - Script repetitive tasks
  - Use open source tools because
    - Aggressively Developed, Keep Pace with State of the Art
    - User-Developed and Instrumented
    - Easy to Assemble into Automated Toolchains

4 Types of Management Tools

- Provisioning
    - Installation of operating systems and other software
- Configuration Management
    - Sets the parameters for servers, can specify installation parameters
- Orchestration/Automation
    - Automate tasks across systems
- Monitoring
    - Records errors and health of IT infrastructure

Toolchain (n):

- A set of tools where the output of one tool becomes the input of another tool

Provisioning activity in order

1. Bootstrapping
    - Cloud Image Launch
        - Use
            - Eucalyptus
            - Openstack
            - Cloud stack
            - abiquo
    - OS install
        - Install
            - Kick start
            - Cobbler
            - Space walk
2. Configuration
    - System Configuration
        - Use
            - BCFG2
            - Cfengine
            - Chef
            - Puppet
3. Command and Control
    - Application Service Orchestration
        - use
            - Capistrano
            - RunDeck
            - Fabric
            - Func