Formulas:

- Float = Latest Finish – Latest Start – Duration
- Expected Time = (Optimum + (4 * Normal) + Pessimistic) / 6
- Variance = Standard Deviation ^ 2 (Standard deviation formulas bellow)
    1. (Pessimistic – Optimistic)/6 (99%) or /3.33 (95%) or /2.6 (90%)
- Z = (T – te)/standard deviation (sd = squar(critical path variances))
- Risk Exposure = Probability * Impact
- Discount factor = (1/(1+%)^year). Present value = value in year /(1 + r)^t
- Discounted cash flow = Discounted factor * cash flow
- NPV = sum of Discounted cash flow Table
- Average annual profit = Net Profit / Number of Years
- ROI = (Average annual profit / Initial Investment) * 100
- Decide between 2 process
    1. Take derivative of each equation 2. Solve for 0 using derivative 3. Plug in found value
- mR = |i – (i + 1)|
- Mean of mR = sum of mR/n-1
- UCL = (mean of mR) * 3.268. LCL = mean of mR – UCL
- UNPL = (mean of mR * 2.660) + Am. LNPL = Am - (mean of mR * 2.660)
- α (standard deviation) = (UNPL – Am)/3
- Cost Variance (CV) = BCWP – ACWP CV% = 100*CV/BCWP
- Schedule Variance (SV) = BCWP – BCWS SV% = 100*SV/BCWS
- Cost Performance Index (CPI) = BCWP/ACWP = 1+CV/ACWP (UB > 1 > OB)
    1. A control chart value indicating closeness to **budget** (SPI budget = schedule)
- Schedule Performance Index (SPI) = BCWP/BCWS = 1 + SV/BCWS (Ahead > 1 > Beh)
- The Complete Performance Index (TCPI) = (BAC – BCWPcum)/(EAC – ACWPcum)
- ECAC = BAC * (ACWP/BCWP) - ETAC = Orig Time * (BCWS/BCWP)
- Critical Ratio(CR) = SPI*CPI (.9<CR<1.2 ok) (.8<CR<.9 or 1.2<CR<1.3 Check) (CR<.8 or a.3<CR Red Flag)
- Channels of communication = (n(n-1))/2
- Based on budge: T = SVc/(BCWSc/TDEV) T=SVc/(BCWSp/TPER
- Based on earned value: T = SVc/(BWCPc/TDEV) T = SVc/(BCWPp/TPER)

Rules:

- Laplace Rule: Neither, equal probabilities
- Maximin: Pessimism
- Maximax: Optimism
- Hurwicz Rule: Blended optimism and pessimism - table of best|worst| a = ??|Blended payoff = a(best) + (1-a)(worst)
- Minimax Regret Rule (Pessimism):
    1. Find highest value in each value
    2. Highest value – cell value
    3. Table with highest value for each row
    4. Minimum value in row has lowest regret
- Zone Rules (if any true then out of control) / x bar is same but mR instead of Am:
    1. A single metrics value lies outside the UNPL.
    2. Two out of three successive metrics values lie more than two standard deviations away from Am
    3. Four out of five successive metrics values lie more than one standard deviation away from Am
    4. Eight consecutive metrics values lie on one side of Am

SLC:

- Waterfall: User only in beginning and end, no need for prototyping, requirements v understood
- V: hard for concur events, no iteration, requitements v understood,
- Spiral: Has iteration, high-med risk, prof concept, r protot, user involved,

- Evo Rapid Protot: v user involved,
- RAD: v user, team familiar with problem and exp, quick product, more peo, need reuse
- Incremental: quick for market window, good for new tech, no iteration with inc, rec und

Terms:

- **Projects**: temporary, defined start and end
- **Operations**: ongoing and repetitive, no end date
- **SDPM** = Software Development Project Management
- **Project Management** covers: Feasibility study → Planning → Execution
- **Project objective** SMART(Specific, Measurable, Achievable, Relevant, Time constratine)
- **Management** = Plan, Organize, Staff, Direct, Monitor, Control, Innovate, Represent
- **Deliverables** = Work Products that will be delivered to client
- **Activities**: Major uni of work, smaller activates or tasks, culminates in milestone
- **Function**: activity or set of act that span duration of project
- **Task**: smallest unit of work
- **Software Process**: a set of interrelated activities, methods, practices, and transformations that people use to develop and maintain software. •
    1. **Software Process <u>Capability</u>**: It describe a range of expected results that can be achieved by following a software process.
    2. **Software Process <u>Maturity</u>**: The extent to which a specific process is explicitly defined, managed, measured, controlled, and effective. It implies a potential growth in capability.
- **Requirements Management**: Establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. This agreement is the basis for planning and managing the software project.
- **Software Project Planning**: Establish reasonable plans for performing the software engineering and for managing the software project. These plans are the necessary foundations for managing software projects.
- **Software Quality Assurance**: Provide management with appropriate visibility into the process being used by the software project and of the product being built.
- The **MOOD** suite of metrics is intended as a complete set that measures the attributes of encapsulation, inheritance, coupling, and polymorphism of a system

Concepts:

- Why do we Measure?
    1. To characterize – to gain understanding of process, products, resources, and environments, and to establish baselines for future assessments
    2. To evaluate – to determine status with respect to plans.
    3. To predict – so that we can plan.
    4. To improve – we gather information to help us identify roadblocks, root causes, inefficiencies, and other opportunities for improving product quality and process performance.
- Capability Maturity Model (CMM)
    1. Initial (Process discipline & Project Management) → 2
    2. Repeatable (Process Definition & Engineering Management) → 3
    3. Defined (Process Control & Quantitative Management) → 4
    4. Managed (Continues process improvement & Change Management) → 5
    5. Optimizing
    1. ROI typically between 4:1 & 8:1
    2. Increases in productivity range within 9-67% and decreases in cycle time range within 15-23%.
- Goal-Question-Metrics (GQM)
    1. Develop set of goals
    2. Develop set of questions to characterize goals
    3. Specify metrics needed to answer questions
    4. Develop mechanisms for data collection & analysis

5. Collect, validate & analyze the data; take corrective action
6. Analyze in a post-mortem fashion, feed forward
7. Provide feedback to stake holders

- Factors That Lead to Inaccuracy:
  - L -- Different Programming Languages You don't know how the translation will affect things
  - C -- Different Levels of Complexity
  - F -- Different Functionality, especially if the new functionality is untried or unknown
  - D -- Different Application Domain
  - A – Different Algorithms
  - R -- Different Level of Reality (Simulation vs. Actual Application; Simulation vs. Emulation)
- Use case points: (sum of actors = uaw) (sum of uc = uucw) (sum of uaw & uucw = uucp)
  - Simple 1 (if system API) – 5 if 1-3 transactions
  - Average 2 (if protocol or real person) – 10 if 4-7 transactions
  - Complex 3 (Interact with GUI) – 15 if more than 8 transactions
- Function points are a software size measure designed to meet three goals:
  1. Gauge delivered functionality in terms users can understand
  2. Independent of technique, technology, and programming language
  3. Give a reliable indication of software size during early design
- The Counting Rules
  - Function point analysis (FPA) quantifies product functionality in terms of five externally visible system elements, called function types, that are readily understood by both users and developers:
  - EI: External input – is a related group of user data or control information that enters the boundary of the application and adds or changes data in an intern
  - EO: External output – is a related group of user data or control information that leaves the boundary of the application.
  - EQ: External query – is a related group of user data or control information that enters the boundary of the application and generates an immediate output of a related group of user data or control information. A query is a set of selection criteria that are used to extract information from an existing database.
  - ILF: Internal logical file – is a user identifiable group of logically related data or control information that (i) resides within the boundary of the application, and (ii) is maintained and used by the application.
  - EIF: External interface file – is a user-identifiable group of logically related data or control information that (i) resides outside of the application boundary, and (ii) is used by the application for some of its processing.
  - Function point counting is a type of Linear Method.
- The Value Adjustment Factor (VAF) can increase or decrease the size by no more than 35%.
- Typical Values for Productivity Rates
  - 50-300 LOC/month (2-15 LOC/day) for high order language
  - 60-500 LOC/month for assembly language
  - Lower numbers are for government projects with very severe constraints, such as embedded real-time systems with life critical safety characteristics
  - Higher numbers are for commercial applications with few constraints
  - Even higher numbers can be achieved if you use 4GLs, high levels of reuse, etc.
- Some Popular Estimating Models (& Tools)
  - Cocomo -- (Costar, Revic, Softcost)
    - (effort = a*size^b) (a=2.4/3/3.6 b=1.05/1.12/1.20) (organic/semidetached/embedded)
  - Putnam's Model -- (SLIM)
  - Shen and Conti -- (COPMO)
  - Lockheed/Martin -- (Price-S)