what is Project Management ?

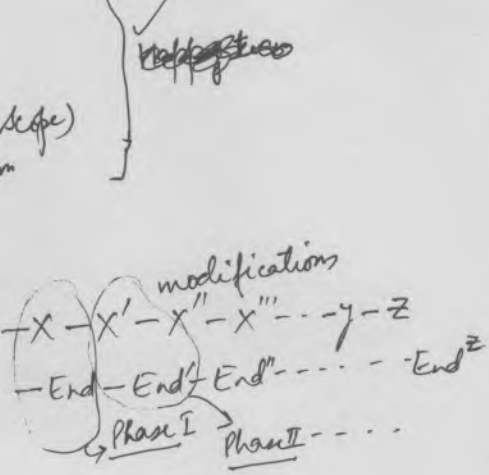
application of Skills & Methods

- ✓ on time
- ✓ on budget
- ✓ to specification (on scope)
- ✓ Happy team

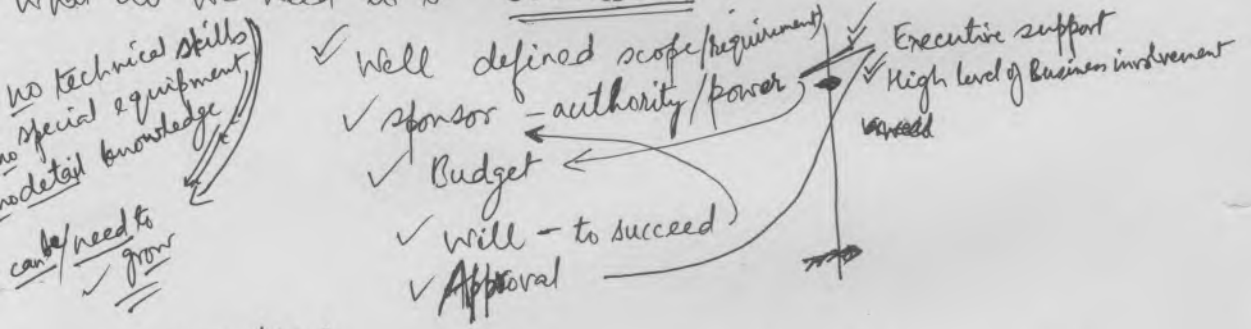
what is a Project ?

PMI
PMBOK

- ✓ Temporary (time-based) duration
- ✓ Defined purpose | specific purpose
- ✓ beginning & Ending
- ✓ Human & Other Resources (Tech. Phy.)



what do we need to be SUCCESSFUL ?



PMI / PMP

Project Management institute → PMP Exam (www.pmi.org)
Project Management Body of Knowledge (PMBOK)
Project Management Professional

How does PMBoK define PM ?

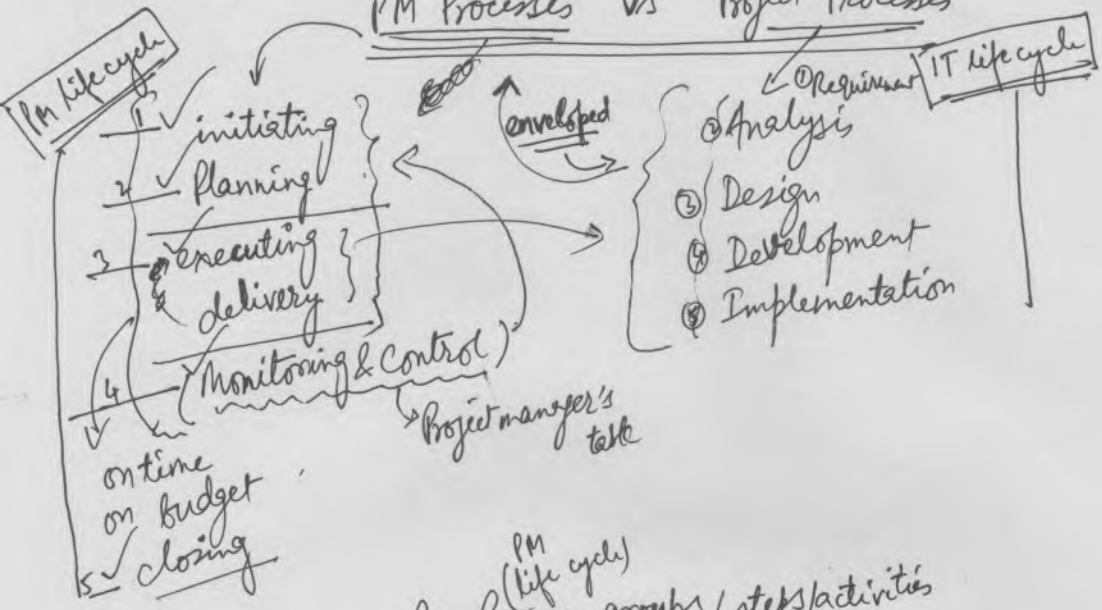
1. Scope
2. Time
3. Cost
4. quality
5. Human Resources
6. Communications
7. Risk ✓
8. Procurement (acquire policies) management

9. Fully integrated → 9 knowledge areas

Project life Cycle (PM)

✓ Project Management
✓ Project Delivery

PM Processes vs Project Processes

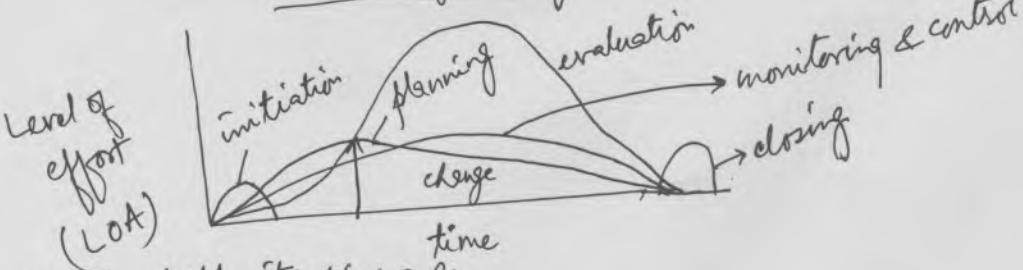


PM (life cycle) Process groups / steps / activities

once / project
once / phase of proj.

- initiating
- Planning
- executing
- monitoring & controlling
- closing

PM Level of Activity



Benefits of multiple iterations of PM

kill points consideration

✓ Revalidation of Project
✓ Replan of project
✓ introduce improvements

Project Organizational structures

- ✓ Frameworks
- ✓ Structures
- ✓ culture

Organizational frameworks

org. chart, Power, Authority → ✓ Structural

✓ Human resource — [how are employees treated, Training Vs work]

Internal, external → ✓ Political

✓ Symbolic → attitude

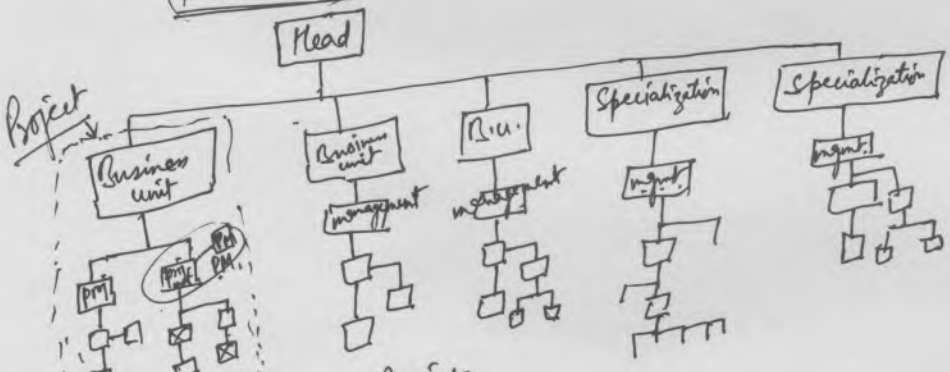
Organizational structures

- ✓ Functional
- ✓ Project
- ✓ matrix (Hybrid)

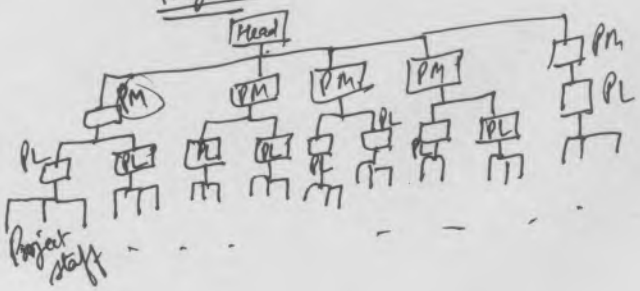
→ variation of these

Power, Authority — Project Success

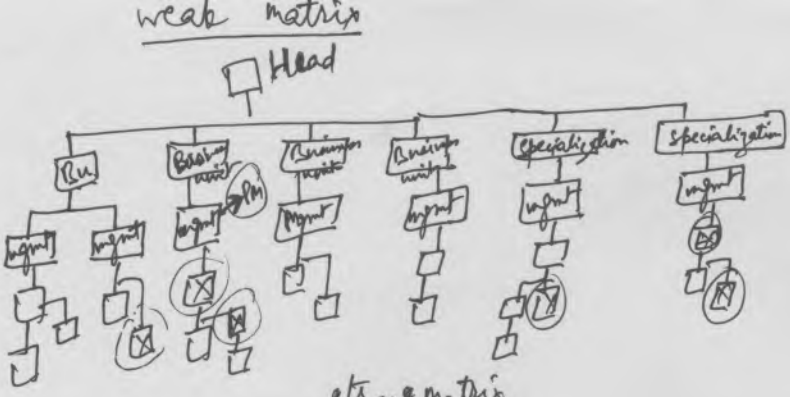
(Functional)



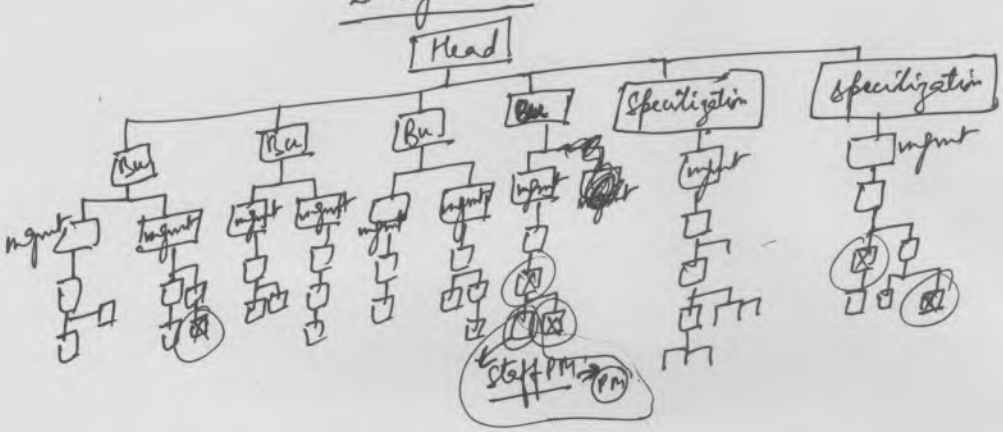
Project



weak matrix



strong matrix



X	PM skills	PM availability	PM authority	Budget ownership	Staff availability	Staff skills
Functional	limited	Part-time	wide range	Functional mgger	Part-time	wide
Project	High	Full	~~Little~~ Full	PM	High	High
weak matrix	limited	Part-time	Low	Functional manager	Part-time	wide
strong matrix	High	Full	medium	PM	some part time some full	wide

organizational culture

- ✓ Identity
- ✓ Corporate vs Individual Emphasis
- ✓ Business Integration – share/compete
- ✓ Risk tolerance
- ✓ Focus – short/long
- ✓ Project/Product/company maturity

Project Scope management

- Scope Planning ✓
  - Level of detail ✓
  - Project environment ✓
  - Scope management plan ✓
- Scope definition ✓
  - deliverables (list of formal deliverables)
  - acceptance criteria — who — which deliverable; How — How do they measure; what — is OK!
- Create WBS ✓
- Scope Verification
- Scope control ✓

(boundaries: what is included / what is not included; requirements; assumption/constraint)

Create work Break down Structure (WBS)

Req def. current situation assessment

- ✓ deliverable based
- ✓ decomposition to work required
- ✓ use a template

- Review system
- interview clients
- understand Faults
- understand needs
- ...

Scope verification –

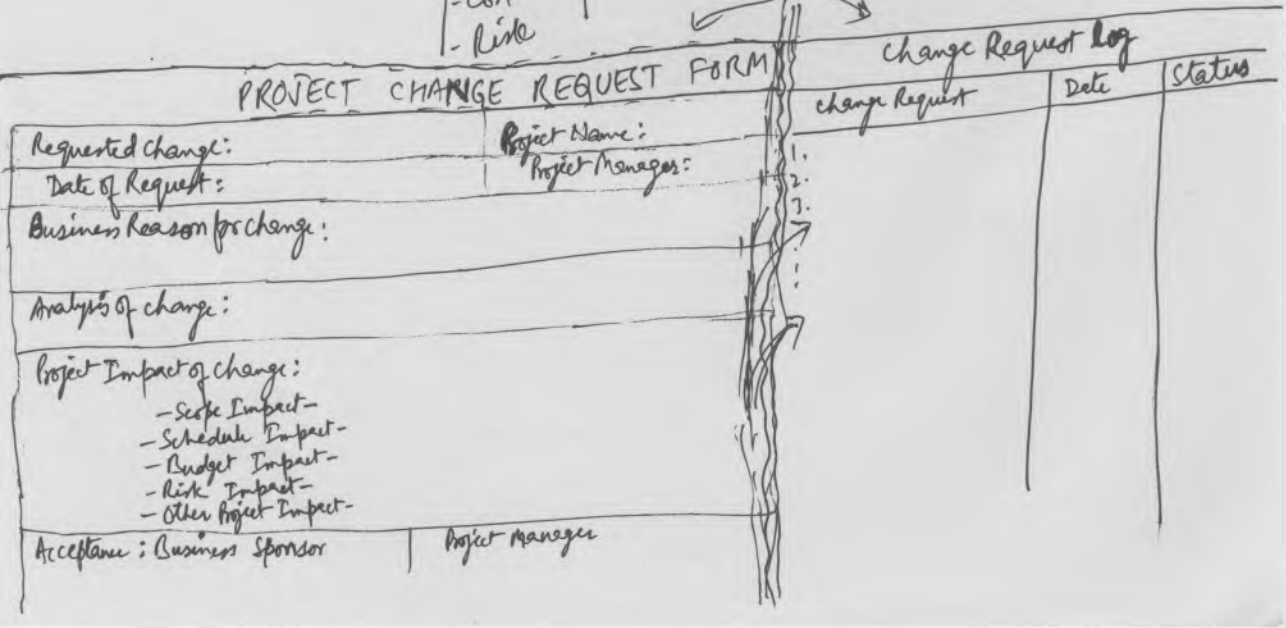
(Formal acceptance of each deliverable/~~work product~~)

- should be straightforward and automatic
  - timeline to review the deliverable
  - turn around process for revision as a single set of comments
  - interim acceptance process
  - resubmit in case of revision/change due to comments

1- ✓
2- -1 ✓
3- ✓
4- x -2 ✓
5- -2 ✓

Scope control =

- Manage change
- Integrate change
  - Schedule | Team
  - cost
  - Risk



Integrate Change

- update plan
- update schedule
- update budget

} change is included ⇒ updated baseline

- Report on change

Minor Change Log

Project Name: Project Manager:
Period Covered:

Change	Schedule Impact	Budget Impact
1.		
2.		
3.		

Total Impact
Approved by:

Project Time Management

concurrent
- Activity definition
- Activity sequencing
- Activity Resource estimating
- Activity duration estimating
- Schedule development
- Schedule control

Activity definition

WBS - Resource
Document work to be performed
work/schedule based
x not deliverable based

Rolling wave planning

analysis
Design
Development
100 Person Hour

Deliverable: D1, D2, D3, D5, D6
Work: T1, T2, T3, T5, T6, T7 — 1 P.M. = 250

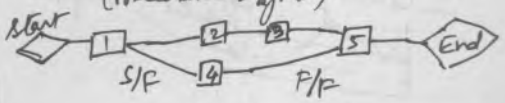
Dependency Types

- Finish-start
- " -Finish
- start-start
- ~~start-Finish~~

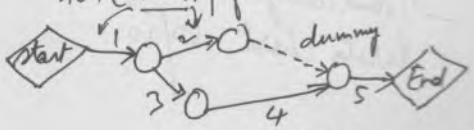
dependency diagrams

- Activity on node (AON)
- Activity on Arrow (AOA)

AON (Activity on Node)
(Precedence Diagram)



AOA (Activity of Arrow)



F/S Finish-start

Activity Resource Estimating

- what resources
- quantities of resources
- when resource needed

Activity Duration Estimating

↳ Amount of work — Team
↳ Actual effort
~~X ~~

Resource availability

- always not 100%

Estimating ?

- Guess ✗
- Parameter driven ●
- multiple estimators
- PERT – multiple Estimates = $\frac{P+4m+O}{6}$
  - pessimistic (P)
  - optimistic (O)
  - most likely (m)

Schedule Development

WBS
Dependencies
Estimates
→ Schedules
↑ ↑ using Software
(Be prepared To iterate)

Critical Path (longest duration)

slack time

Resource Leveling

- optimal teamsize (Cocomo)
- Full utilization

= Schedule Control = (managing schedule)

- weekly Statistics
  - by task
  - How much time this week
  - how much remaining time
- Track Progress
  - Integrate changes → to Forecast/finish
  - manage against baseline
- → Update plan – Actuals/ETC into software

✓ Schedule Compression

critical path tasks
- Fast Track → change dependencies?
  → add resources!
- Crash
  → change environment
  change tools
  change estimates

Project Cost Management

- Cost estimating
- cost budgeting
- Cost control

= Cost Estimating = cost for each WBS (work break down structure) element

✓ Team $
✓ Materials $
✓ Supplies $

- Life cycle costing
  examining trade off – spending = more now (analysis & design)
  → spending less later (coding & testing)

Cost elements

direct costs – salary
Indirect costs – fringe benefits, other costs

Fixed costs – only once
Variable costs –

Cost variability

change over time
change over volume

Estimating techniques

- analogous –
- unit cost based on effort
- Parametric
- Vendor Bids / Fixed price

Cost estimates – WBS element

Cost Budgeting

- Aggregate costs – All (direct/indirect/Fixed/variable) cost
- cash flow
- cost Baseline
- Contingency unknown

Project Budget Accounting

Present value (PV) (FV)
net " " (NPV) - Return
Rate of Return (RoR)
Payback Period-(Length)

= Cost Control (cost management)

- manage cost Baseline → -Approve all changes
  -Manage changes
  -Approved changes
  -Estimate variances
  -Price variances
- manage variances
- managing Budget performance

Managing Budget Performance

Budget - Schedule

on Time - on Budget

✓ Ahead - over
✓ Ahead - under +
✓ Behind - over −
✓ Behind - under

(Earned Value analysis)

Microsoft Project Professional 2003

Tools/options

view | calendar | Calculation | Schedule

(Fixed work)

default task type:

Project Quality Management

- Quality planning < define quality standards / Build quality in your project
- Perform Quality Assurance
- Perform Quality Control

what is Quality? = ✓Conformance to requirements (delivered each & every component)
— ✓Fitness for use

✓(Customer's satisfaction)

Relationship to other quality standards:

ISO	Joran
TQM	Crosby
Six Sigma	Deming

✓Continuous Improvement

✓Prevention vs Inspection → (early detection)

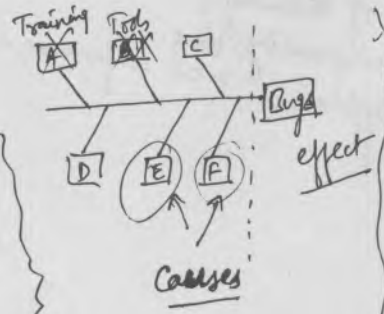
Quality Assurance: {— doing the right things / — doing things better}

Root cause analysis

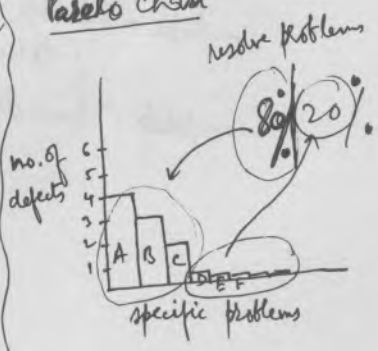
Quality Control: (producing the right results – Fitness for use)

- ✓ Testing
- ✓ Inspection
- ✓ sampling
- ✓ Statistical control

Cause & Effect Diagram:
(fish-Bone diagram)



Pareto Chart



Procurement plan development

Procurement planning (Purchase)
- make/buy
- How to buy
- what to buy
- How much to buy
- when to buy

Purchase planning
- Request for proposal / RFI / RFQ
- list of requirements
- evaluation criteria
- clarifications & questions

Vendor selection
- score the responses
- vendor shortlist
- presentations / Pilot project
(Best response)

Contract administration
- contract type (fixed price, cost plus, per diem (unit of work/...))

Contract closing

Statement of work
- project plan within project plan
  - scope
  - time
  - cost
  - → contract

Project weekly status Report

Project Name:
Reported by:
Reporting Period From: To:

what I was able to accomplish this week:

what I had planned to accomplish but was unable to:

Unplanned work Accomplished

what I plan to do next week

Problems/Issues/Warnings

Project Monthly Status Report

Project Name:
Reporting Period:
Prepared by:

Major Accomplishments this month:
* Accomplishment 1
* Accomplish 2

Financial Status:
Total Project Budget
Spending to Date
Estimate to Complete
Variance

Project Schedule:
Effort Expended to Date
Remaining Effort
% Complete

Major Issues:
* Issue 1
* Issue 2

create Project Directory (LAN Directory)

- Project Directory
  - Action Items
  - change Requests
  - Decision Requests
  - Meeting Minutes
  - Monthly status reports
    - April
    - August
    - Feb
    - Jan
  - Problem Reports
  - Project Plans
  - Weekely Status Reports

web Sales websites

Binders

✓ Meetings ✓

* — Agenda (objective)
— Attendees (right people)
— Followups / Action Items
— on time
— minutes (publish conclusions of meeting)

= Stakeholder management = { — communicate; — Followup/Issue log (Issue management log); — consistency }

Issue management log ←

Project Name:

	Issue Name	Date Opened	Due Date	Current Status
1.				
2.				
:				

= Project Risk Identification =

Risk = event/component that involve uncertainty/unplanned event (unknowns) | (unknown risk, known risk)

-/+

Risk management → to change probability of risk
" impact " "

Risk Management planning

- Conduct stakeholder analysis (tendency — risk seeker? risk avoidance?)
- Corporate Factors
- History/Experience of organizatin to deal with risk (Risk profile) — Appropriate Processes // time // money (cost) $

Risk Probabilities and Impact Assessment

multiply X	Low(1)	Medium(3)	High(7)	Very High(9)	
Cost(2)	Insignificant	○ <5%	○ <10%	○ >10%	○
Time(3)	Insignificant	○ <10 days	○ <25 days	○ >25 days	○
Scope(5)	Insignificant	○ No business impact	○ Some business impact	○ definite business impact	○
Quality(2)	Insignificant	○ No anticipated problems	○ Potential for problems	○ definite problems	○

Result Low

Ranking High > 75
Medium > 50
Low < 50

Risk Management Plan

Project Name:

1. ~~Executive Summary~~
2. Risk ~~mgt~~

1. Executive Summary
2. Risk Management Approach & strategy for dealing with risk/How you are going to dealing with risk)
3. Roles and Responsibilities (who is directly responsible for dealing with risk)
4. Risk categories & categorize risk (are by technology, HR, business, -)
5. Risk Probabilities and Impact
6. Ongoing Risk Management Process (how to manage risk by all your processes in your project)

Sources of (~~known~~) risk

- Technology
- Team (training related, HR related, ...)
- Scope (
- Business (growth mode? stable mode? decline mode? rapid growth?)
- Economy (poor economy? ...)
- Competition (your company ~~having~~ leading more risk taking?)

Risk identification Techniques

- ~~Brainstorming~~ - Brainstorming
- Delphi technique (finding known experts in the field, interviewing them and ask for help)
- interviews (stakeholder, ~~known experts~~, ...)
- Root cause analysis of (Delphi technique, interviews ..)
- SWOT analysis (
- Experience

= Risk Register =

Project Name:

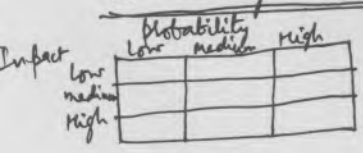
Risk Item	Probability/ impact	Risk management Strategy	Critical date	current Status
inadequate training	medium			

Project Risk analysis (

- qualitative analysis → Probability/Impacts → put in impact matrix → timeline
- quantitative analysis
- response planning

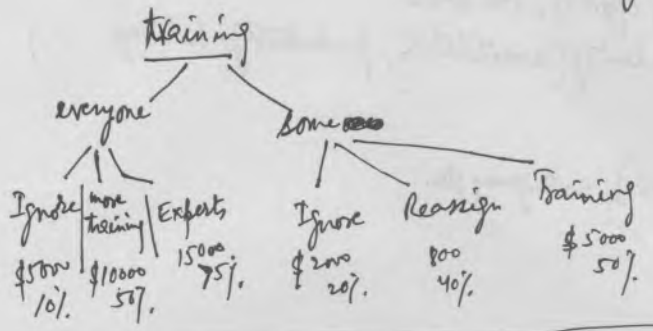
(Focus on the right risks/Solve high risks, ignore low risks)

Risk analysis matrix:



Quantitative Risk analysis :

= Reserved for high risks
- numeric rating → $ / time
- ✓ monte Carlo simulation (by running simulation with probability vs Impact ⇒ probability distribution)
- decision tree (break risks/events into a number of subsequent actions and assign and determine the value of risks associated with the branches of tree)

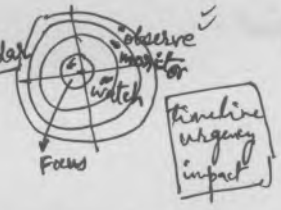


Risk Response Planning

- Avoid → by changing plan
- Transfer → by buying insurance, contract with outside agency
- mitigate → lessen/minimize the impact
- accept

Project Risk Management

= Monitoring and control ✓
- ✓ Monitor known risks ⇒ examine risk register ⇒ project timeline
- ✓ watch for new risks
- ✓ manage contingency (future uncertain events)



watch for new risks that comes from:
- scope changes
- Schedule changes
- environment changes
- Ask your team

known Risks		unknown risks	unplanned deviations
Risk 1	5 days $5K	General contingency	known Risks risk 1 5 days 5K
Risk 2	10 days $15K		unknown 20 days 25K
⋮	⋮		
Risk 15	8 ..	$12K	

✓✓ * reserve funds for risk management

Proposal Evaluation sheet

Company Name:
Evaluated by:

1. References:
   Rate 1-5 (5-Exceptional, 4-very good, 3-Average, 2-Below Average, 1-Below Average)

	Rating	weight	Score	Cost
Reference #1				
Experience with package	3	2	6.00	
Experience with company	2	2	0.10	
Experience with implementation				
Experience with Post Implementation Support				
General Impression				
Reference #2				
			0.50	

Total Vendor References

2. Suitability of the proposed product:
   Rate 0-5 (5-exceeds, exceptions, 4-fully meets, 3-meets but will require workarounds/procedure changes, 2-partially meets, 1-very incomplete solution, 0-Does not meet)

General Functional Requirements	Rating	Priority	Score	Cost
Accident/incident Reports				
Claims				
measurement Equipment/Life Safety Equipment				
Document Repository and change management				
Legislative and Operational Requirements				
Audits				
Risk assessment/management				
Routine maintenance and Inspections				
Casual Analysis/Corrective Actions				
Lessons Learned and followups				
Interface Requirements				
Technical Requirements				
			0.00	

Total suitability of product	Rating	Importance	Score	Cost
3. Customer Support and vendor stability				
Rate 0-5 (- - - - - -)				
Technical support				
vendor Response Times				

4. - - -

Proposal Evaluation Sheet (Summary)

Company name :
evaluated by :
General comments

	10	0 5 0
1. References		
2. Suitability of the proposed product		
3. Customer support and vendor stability		
4. Industry best practices		
5. implementation plan		
6. Cost of proposal		

Earned Value analysis (Tool kit)

Combines ✓Scope, ✓Time, ✓Budget } Total project performance

Earned Value Terms:

- ✓ PV – Planned value → original baseline
- ✓ EV – Earned value → what was actually done.
- ✓ AC – Actual Cost → what it really cost
- BAC – Budget at Completion → total budget
- EAC – Estimate at completion → current forecast
- ETC – Estimate to complete → remaining work
- VAC – variance at completion → variance

Earned Value calculation:

Cost variance, CV = EV − AC
Schedule " , SV = EV − PV
Cost performance Indicator, CPI = EV/AC
Schedule " , SPI = EV/PV
Estimate at Completion, EAC = BAC/CPI
Schedule " " , SAC = original date/SPI

CV = $3800 − $5000
= − $1200
SV = $3800 − $4500 = − $700
CPI = 3800/5000 = .76
SPI = 3800/4500 = .84
EAC = $10000/.76 = 13137 $
SAC = 6 month/0.84 = 7.1 month

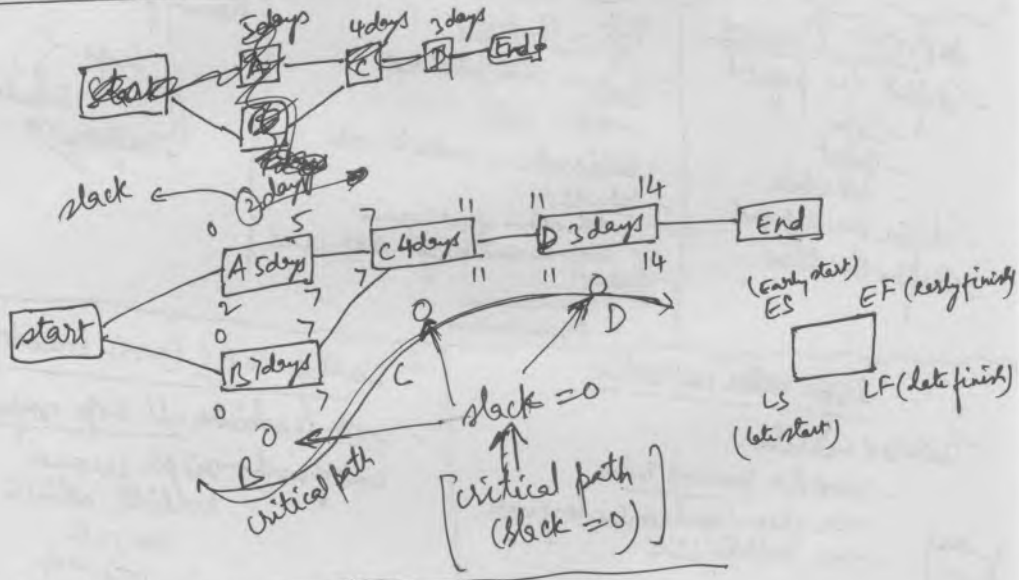
Creating network diagrams

- defines the project schedule
- identifies the critical path
- supports fast track / Crashing

How to create network diagrams :

Step 1- draw basic diagram with ~~dependencies~~ dependencies
" 2- Add effort (e.g.; in days)
" 3- forward pass - Early start (ES)
Early finish (EF)
" 4- Backword pass - late start (LS)
late finish (LF)
" 5- Calculate slack ✓
Identify critical path ✓

Step 1 -

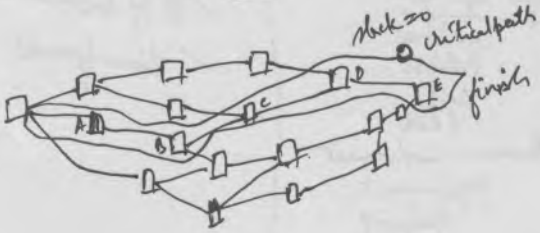


---

types of slacks

1 ——— (task) Free slack (float)

2 ——— (project) Total slack (float) = latest possible start date that any component to the project can start without impacting availability to finish the project on time)



PM processes

① – Project Initiation and Scope definition
= identify project objectives
= develop project charter
– develop preliminary scope statement

Initiating

= define the business need/req. | = define Technical requirements
– identify the sponsor
– assign the project manager
= identify stakeholders
– justify the project

(i) Project charter (detailed report) high level

High level Business case + Technical

– Introduction
– Business objective (requirements what?)
– preliminary cost/Benefits

Problems
Solutions
Benefits
(approval) signed by —

(ii) Preliminary scope statement (in scope not detailed like if — then — doc ---- programming)
describe business rules

(approval) signed by —

– Current situation
= project requirement — Scope (low level info!)
– project schedule — Phase1 – 6 months, Phase2 – 9 months, ...
= assumptions/constraints (impediment)
– Risks assessment (constraints)

(develop PM plan)

② Project planning

= Scope planning — what is in scope & what is NOT in scope
– create WBS — identify all deliverables/all packages — Req.doc... + design doc + project scope
= define schedule — combine [WBS + estimates + resources] → realistic, balanced, full utilization, mildly agressive
= define develop budget — $ Resources, $ materials, $ services, $ contracts
– define quality
– develop HR plan
✓ – Identify Risk
– plan procurement

= define approach (Agile, Rapid prototype, RUP, ...)
– define team (skills, resources, time, ...)
– understand business motivation

③ Project executing
– direct and manage project
– perform quality assurance
– acquire project team
– develop project team
– information distribution (communicate)
– request seller Responses
– select sellers

④ Project monitoring & control
– integrated change control
– scope verification
– control (scope, $, Time)
– HR management
– performance reporting
– risk management
– contract administration

⑤ project closing
– close the project
– contract closure
– project review (lessons learned, achieving best practices)
– project handover (training, operation, support, maintenance)
– project acceptance (acceptance document, warranty support, conditional acceptance)
– project support

What is project management ?

- on time
- on budget
- on scope
- Happy team

What is a project ? (for IT)

PMI
PMBOK
- Temporary
- Specific purpose
- Beginning and end
- requires resources (HR/Tech/Physical)

What is PM ? (PMBOK guide)

fully integrated
- Scope
- time
- cost
- Quality
- Human Resources
- Communications
- Risk
- Procurements

Project management life cycle

- Initiating
- Planning
- executing
- monitoring & controlling
- Closing

IT Project life cycle

- Requirement analysis — Initiating, Planning
- Design — executing, monitoring & control
- Development — executing, monitoring & control
- Implementation — executing, monitoring & control → closing

Why PM ?

+ 30%, total failure
+ 30%, (challenged) Problem projects (over cost, over schedule, poor/limited solution)

Standish Report

Reasons :- define requirements, lost data, changing technology, often cross organizational departments, less mature IT industry

PM Success Factors

- executive support
- High level of business involvement
- well defined requirements
- well PM best principles practices

Project initiation and Scope definition

consensus approval

- Identify Project objectives
- Develop Project charter
- Define Preliminary Business Requirements
- Define technical requirements
- Identify Project stakeholders
- Develop Project Scope

Identify Project objectives and Requirements

- understand source of Request
  - Business Plan
  - Regulatory compliance
  - marketing
  - competition
- understand position of requester
- Validate feasibility of Request

* High level definition
* High level Justification
* Business case

for approval
Business expert
IT expert

Project charter

Project charter
Project X

Date: YYYY/MM/DD
Prepared by: Gufran Ahmad

Table of content

1.0 executive summary
2.0 Project sponsor and key stakeholders
3.0 Business case → (problem) Why need project
4.0 Objectives and Success criteria
5.0 High level cost/Benefit Analysis
6.0 Risk ~~analysis~~ assessment.
7.0 assumptions and constraints

Project Scope statement

Project Scope statement
Project X

Date: YYYY/MM/DD
Prepared by: Gufran Ahmad

Table of content

1.0 executive summary
2.0 Project Goals
3.0 Project Deliverables
4.0 Project Scope/Boundary
5.0 Success Criteria
6.0 Cost Benefits Statement
7.0 Project schedule
8.0 Project Budget
9.0 Risks
10.0 Assumptions and constraints

define Business Requirements

- High level analysis
- Gather Requirements
- validate assumptions
- Identify Business drivers
- Identify "speed bumps"/impediments

define Technical Requirements

- Availability / Response Times / Performance
- existing investement
- Proposed Direction

Stakeholders

- Project ~~stake~~sponsor
- Business users
- other departments
- Customers / suppliers / vendors
- Project team
- opponents

Develop Project Scope statement

- Formal definition of what
- Formal approval

Obtain Consensus and final approval

- Presentations
- workshops
- Approval

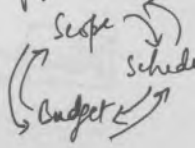
~~Project~~ Planning

- define the project
  - Scope
  - Buget
  - Schedule
- define the approach
- define the team

Define the project

- Scope – what is/is NOT
  - work – WBS
  - Deliverables – make it work
- Schedule
  - estimates – effort required
  - Resource availability – staff/team data
- Budget
  - Human resources
  - Physical resources

Planning process



Scope ~~data~~ narrative

- Detailed narrative
  - Describe business rules
  - use clear/~~and~~concise language
  - use bullets/lists

Develop WBS (work breakdown structure)

- work to achieve all scope components

unit of work – single resource
realistic estimate
40-80 hr
1 – 2 weeks

Project Deliverables

- Formal deliverables
- Informal deliverables

Project approach

- Project life cycle
  - traditional
  - Prototype
  - Agile
  - RAD
- Business involvement
  - Requirements
  - training
  - testing
  - documentation

define the team

- Skill Requirements
- # of resources
- time frames

Project strategy development and ~~planning~~ preliminary planning

- understand Project motivations – (Business motivations, technical motivations)
  - new/extend functionality
  - Reduce labor
  - Reduce costs
  - Improve service
- define the requirements
  - Business functions
  - Technical "
  - performance requirements
  - understand constraints
- Project considerations
  - Success factors
  - alternatives
  - organizational resources
  - technical resources
- acceptance Criteria
  - Clear and measurable
    - Acceptance test
    - no critical problem
    - <5 faults
    - <20 issues
  - Predefined
  - ~~Ob~~ obtainable
- Risk assessments
  - understand the source of risk (business, competition, technology, resource
  - counteract the risk (estimate, mitigate, transfer, avoid, deny, acceptance)
- develop preliminary approach
  - Schedule
  - Budget
- Preliminary plan
  - statement of requirements
  - Benefits statement
  - Preliminary budget
  - Preliminary schedule
  - Risk assessment

Requirements analysis

- Document Project (Business) Requirements
- Define target solutions
  - understand current situation (current business, current system . . .)
  - understand desired state (business functional requirements, Goals, performance, availability, tech, environment)
  - Define conceptual solution
- Review alternatives
  - manual vs automated
  - Build Vs Buy
  - Renovate vs Replace
  - Internal Vs external
- Define target (Conceptual) Solution
  - document solution (business, technical . .)
  - identify risks
  - document shortcomings
  - document assumptions / constraints

Creating WBS

- document the work
  - Based on scope (start with scope, . .)
  - produces the deliverables (define major + minor deliverables)
    (identify work packages → decompose work packages)
- Building block for project planning
- most detailed view of project

How to develop a WBS?

- use guidelines
  - Previous Projects
  - Templates
  - Industry models
- use analogies (similarity . . .)
  - Top down / Bottom up
  - Include milestones

Tips & Hints

- involve the team
- involve the Business | use resource have support
- Consult experts
- sticky notes on white boards
- Complete one layer at a time
- don't worry about sequence
- don't worry about estimates

Estimation

✓ Realistic estimates
- adjust for reality
- resource availability (not 100%)
- adjusting for skills (normal/average, variance from actual assignments

$$PERT = \frac{4\,Realistic + Optimistic + Pessimistic}{6}$$

✓ Actual effort
- effort ≠ duration (effort is work based
- effort is independent of (availability, productivity, holidays ...)

✓ issues with estimates
- dealing in unknowns
- poorly defined requirements } contingency plan
- changes

✓ Between 40-80 hours

✓ Sources for estimates
- analogous
- expert judgement
- quantitative
- parametric (applying ratio for analysis, design, dev. ...)
  ~10%, ~25%, ~45%

~~Adjust~~.

Schedule Development

✓ Define key project data

✓ Identifies resource requirements

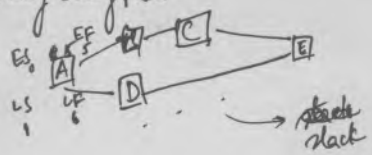
✓ Creates critical paths & management
- determine project end date
- " " critical task
- manage near critical path/task ✓ (small slack)

✓ How to create a schedule

Iterative

WBS elements
Task estimates
Resource assignments (link resources to tasks based on availability, resource calendar/matrix)
Task dependencies (mandatory, discretionary, ...)

✓ dependency diagram



— Budget development

✓ identify all project costs (HR, infrastructure, supplies, overhead ...)

✓ Allocate contingency (Project contingency, management reserve) ~~based on risks~~

✓ create project budget — time line (not a single number, allocate over the time, cash flow management, work with finance department ...)

✓ create project baseline (project budget)
- approved budget
- used for mgmt. (Finance, variance reporting)

✓ estimating process —
- analogous (similar project ....)
- parametric (based on ratio/proportion ~% ~% ~%)
- definitive (bottom up)

esteem: use expert, use experience

— Quality Management Plan Development

✓ Define quality requirements
- acceptance criteria (bug free, no errors ....)
- Quality measures
- quality Cost/Benefit analysis (Prevention, Appraisal, Failure) Value added

✓ identify quality measures
- Inspections
- walkthrough
- Reviews
- Testing (unit, system, acceptance, ...)
- independent reviewers

Preventative actions / Right product / Right things

✓ create quality management plan (Formal definition of quality, assigns responsibilities, obtain approval/commitment)

— Build Quality In
- Put quality tasks in WBS ✓
- allocate adequate time for quality
- protect quality tasks

✓ dispute resolution
- predefined approach
- identified participants
- non threatening

Risk management plan development

✓ Risk profile (adverse, neutral, seeking)
like R&D

✓ identify risks
- Source of risk (technology, HR, Legislative, business, competition, ...)
- Brain storm (involve team, ...)
- Prior experience

✓ Analyze risks
- Review & understand risk
- understand the probability
- understand the impact
- determine relevance
- risk exposure = P × Impact

✓ Strategize risks
- ~~risk exposure = P × Impact (measures)~~
  - avoid
  - transfer
  - mitigate
  - accept

✓ control risks
- contingency

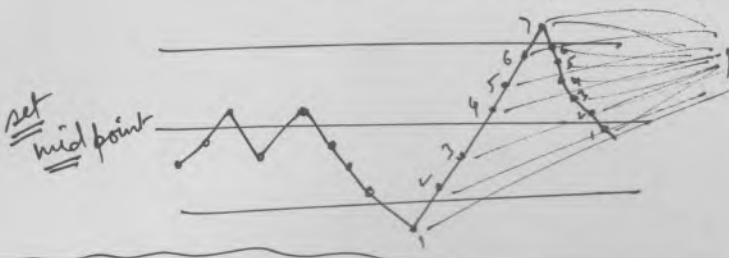
✓ contingency plan
- allowance for risk
  - task specific
  - project general
- Build into the project

✓ Advance risk mgmt. —
- timeline analysis
- Quantitative methods
  - × monte Carlo
  - × Simulations
  - × decision trees

Communication planning

- what to communicate (detailed status, summary status, changes, issues, success, - - -)
- when to communicate (constantly, daily, weekly, monthly, quarterly, milestones - - -)
- who to communicate to (team, acceptor/sponsor, senior management, all stakeholders, external vendor/customer)
- How to communicate (Formal (written, presentation, email) / informal (verbal, email)) / Project web site
- no. of communication channel = $\frac{n*(n-1)}{2}$
- meetings (agenda, only necessary attenders, start/end on time, follow up/action items)

Control chart for Quality



Rule of 7 points

~~recognize~~

7 consecutive points (measurements) going in the same direction,

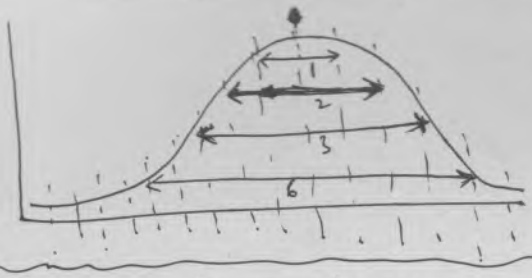
⇓

potential tolerant situation

so, corrective actions must be taken.

(by reverse process)

Six Sigma quality standard tools



accuracy

1 sigma → 68.26 % (defect-less)
2 " → 95.46 %
3 " → 99.73 %
6 sigma → 99.99 %

Project Human Resource Management

- Human Resource Planning ← develop project organization chart, define roles and responsibilities, develop staffing plan
- Acquire Project Team
- Develop Project Team
- Manage Project Team

Introvert / Extrovert (Soft skills)

Hard skills

Motivator
Thinker
Director

Communication management

- planning
- Info. distribution
- Performance Reporting
- Manage stakeholders

right people / right information / right method format / right time

who what How when

- status (reports, presentations) (project websites)
- plan updates
- team performance
- meetings
- problems

team building

✓ Acquire the team
- create staffing plan (HR, skills, timeframes)
- interview potential team members
- create individual project contracts
- staff the project
- conduct team orientation

✓ Determine training requirements
- technical
- managerial
- Business
- communications
- develop training plan

✓ Adjust the plan
- add resources or time } junior resources
- ensure mentoring exists
- consider reducing time } senior resources
- consider how to challenge

✓ Roles & responsibilities
- project organization chart
- clearly define management structure
- define reporting relationships
- clearly define ALL project roles

✓ standards & procedures
- team norms (communications, work hours, availability, quiet time ↳(focused time))
- project (documentation, programming)

✓ distributed teams
- require extra work
- take advantage of technology (teleconferencing, web conference, instant messages)
- respect time differences

— Presenting the ~~complete~~ comprehensive project plan —

✓ consolidates all planning results

✓ presents project plan for approval

✓ initiates controlled updates of plan due to changes
- continuous maintenance required
- controlled maintenance required

✓ Project plan content
- admin details
  - TOC
  - Revision history
- executive summary
- sponsors / stakeholders
- project organization
- scope definition
- assumptions & constraints
- schedule
- budget
- quality
- risk
- procurement (purchase)
- milestones

✓ Present plan for approval to
- project sponsor/acceptor
- IT management
- Business mgmt.
- HR mgmt.
- Procurement mgmt.
- quality mgmt.

circulate plan widely

Project execution, control and coordination

Project execution — to delivery
- execute WBS
- delivery the project
- manage procurement
- eliminate road block, facilitate team performance by ensuring access to resources, effect decisions, secure commitment
  - ↳ key. def. doc
  - → approval

Project control — to delivery on time, on budget
- manage scope
- " quality
- " team
- " risks

| may replan (adjust plan based on facts, reassign resources/tasks, reschedule, replan, adjust

Coordination between team - business - project - input - project - stakeholders
- keep all stakeholders informed
- manage changes
- manage decisions
- communicate

Change Control

- Integrated change management
  - change control form
  - change control log
  - change impacts (scope, cost, time, quality, human resources, procurement, risk) request
    * require communication

Project change Control Log

change #	Create Date	Resolution Date	change Request Name	status

- Scope Creep
  - small incremental change by Running log
  - because of: Open ended definitions, Slow decision, poorly written scope/project plan, Refined Business understanding/Requirements, Innovative developers

- Change control Board
  - formal change authorization (organizational)
  - meet both project & organizational (priorities & requirements)

project change request form

Project name:
Project Manager:
Request Name:

Date issued:
Due Date:
Request #:
Prepared by:

Reason for change:

Description of change:

Prepared by:

Cost Amount:

Prepared by:

Ramifications (e.g.; schedule and staffing):

Prepared by:

Approved:
IT Manager
Name:
Signature:
Date:

Approved:
Project Authority:
Name:
Signature:
Date:

Quality for management

- Quality assurance
  - meeting project specifications
  - adhering to standards & Procedures
  - Doing the right things
- Quality Control
  - testing and validation
  - Doing it right
- testing
  - (unit, system, factory, performance, & acceptance tests)
    - unit → bugs
    - system → file, interface, OS
    - factory → Implementation tests at development site
    - performance → volume, stress, response
    - acceptance tests → Business scenario
- Rework & Retest
  - document test failures
  - Correct defect
  - Rerun sufficient tests to revalidate
  - + Regression test
- What to test
  - test more than SW
  - documentation
  - user manuals
  - training manuals
  - operational manuals
  - } meet final system spec.
- Quality measures –
  - control chart
    - upper
    - Expected
    - lower
    - out of control
  - sampling (randomly)
  - Trend analysis – rule of 5 sample points
  - Pareto chart 80/20

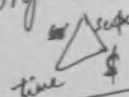
team management

- Motivation (team motivation, individual motivation, challenging assignments, meeting career plans, helping team)
- Conflict resolution (team members, project & Business, project & vendor, immediate, non-threatening, positive, non-personal)
- Recognition Programs (team & individual, meet personality profile, work for the group)
- Performance reviews (be specific, be fair to +k/-, ~~no surprises~~ no surprises)

team development (Forming, storming, norming, performing)

Resource management and coordination

- ~~Optimize~~ optimize team performance (match skills to tasks, match performance to schedule, match expectations to opportunities)
- optimize project performance (deliver the project smarter, use better tools, reuse don't invent, work smarter not harder)
- problem resolution (workshops, decision trees, root cause analysis)
- optimize the project model (Rationalize the project model)



CMMI – (Capability Maturity Model Integration)

* process improvement approach
* consisting three parts :- ① CMMI development (22 process areas)
② CMMI Services
③ CMMI Acquisition

* 22 process areas :- (has 1 to 4 goal ; each goal is comprised of practices) → specific goals and practices, in addition to generic goals & practices

- CMMI Casual Analysis and resolution
- CMMI Configuration mgmt.
- CMMI decision analysis and resolution
- CMMI Integrated Project mgmt.
- CMMI measurement and analysis
- CMMI organizational innovation and deployment
- CMMI Organizational process definition
- CMMI " focus
- CMMI " performance
- CMMI " training
- " product integration
- " project monitoring and control
- " planning
- " process and product quality Assurance
- " quantitative project mgmt.
- " Requirements development
- " mgmt.
- " Risk mgmt.
- " Supplier agreement mgmt.
- " technical solution
- " validation
- " verification

---

* types of appraisal (evaluation) = A, B, C (class)
  - Appraisal Requirement for CMMI (ARC) methods
    - Standard CMMI Appraisal Method for Process Improvement (SCAMPI) for class A
    - Process Professional Assessment (PPA) for class A, B, C

* two different representations – staged and continuous

- staged model: as in CMM uses (5 maturity levels process areas) to measure process improvement
- continuous: capability levels within each profile as in system-engg. CMM

CMMI

(Capability Maturity Model Integration)

Benefits of Integrated Process Improvement:-

- cost benefits (in training multiple models & appraisal method, maintaining redundant process assets in a repository, maintaining or purchasing expertise in multiple models)
- Clarity of focus (unify and reinforce vision, efficiently marshal and apply scarce resources, provide common language across various disciplines)
- Process integration and lean organization (increased value to the customer by waste elimination in production of product)
- flexibility and extension into New disciplines (ability to add disciplines as the business or engineering environment changes without affecting fundamental process improvement structure and terminology)

Implementing Integrated Process Improvement:

Principles:

1. Maintain executive support (Concentrate on keeping your executives aware, involved, and excited)
2. Pick your targets carefully (don't underestimate the effort needed, because process improvement is hard work)
3. Leverage best practices (use what is available and working and build on those methods)
4. Align process improvement with business objectives (use existing objectives or create new ones to support process improvement, advantage of)

Practical Advices:-

- explicitly identify, recognize, and map legacy process-improvement investments to the integrated effort so they will not be marginalized or duplicated.
- select and tailor the model to the business. Use local language to describe activities rather than "model-ese".
- an integrated process-improvement group is essential to success.
- enthusiasm is a plus, but manage "experts and zealots" carefully - they can sometimes raise more barriers than they overcome.
- Integrate process-improvement reviews into project management reviews.
- capture the hearts of middle managers.
- make sure that implementation is as strong and coherent as the vision.
- train to organizational processes rather than to models
- Continuous appraisal of projects emphasizes proactive process improvement, replaces the audit philosophy, and decreases the fear factor.
- maintain constent appraisal team for consistent findings.
- Process teams of real process users can force the process definitions to be user-friendly and address real-world multidiscipline concerns.
- A well-designed cost accounting system facilitates the collection of useful process metrics data, particularly for cross functional activities.
- make a focused, continuing effort to identify common process opportunities.
- using "workshop" approach to process engineering can result in a tenfold increase in productivity for typical meetings.
- The early use of pilot projects with a draft version of an integrated process allows for testing and fine-tuning of the process. A good cross-section of projects gives early feedback on where improvements are needed.

- don't try to integrate disciplines that differs widely in their process maturity.
- "steal with pride" is better motto.
- focus on the development of a straightforward and consistent method for tailoring standard processes, especially for organizations that include highly diverse project domains.
- Remember that any change in an integrated environment will probably affect at least one other group.

CMMI Concept

- Process Content
- Process improvement
- CMMI & Business Objectives = produce quality products or services (by requirement management, quality assurance, varification & validation)
  - create value for the stakeholders
  - be an employer of choice
  - enhance customer satisfaction
  - increase ~~sth~~ market share
  - implement cost ~~sav~~ savings and best practices
  - gain an industry-wide ~~rec~~ recognition for excellence

PMP- Project Risk management

6 steps process:

① Risk management planning = how to approach and plan risk management activity for the project
② Risk identification = which risk might affect the project and documenting the characteristics of those risks
③ qualitative risk analysis = prioritize which risks we need to address
④ quantitative risk analysis = putting numbers to risks, trying to identify which risks are key to the project that we need to address
⑤ risk response planning = response plan to either enhancing opportunity or reducing risk threats
⑥ risk monitoring and control = ongoing assessment during entire project life cycle

Risk := uncertain event or condition that if it occurs, causes positive or negative effect on your project.
(whole notion about opportunity)

= known and unknown risks :=

unknown risk ⇒ e.g. (financial, rate of inflation that may exist in particular country where we are doing project)
- comes from the areas when we did not consider

Risk management intension — to reduce the level of uncertainty

* in the beginning of project, the level of uncertainty is the highest. (

* there are two basic types of risks (business risk and pure risk)
(opportunity to gain or loss) (only contains opportunity to loss)
(insurable risk): transferrable to..

why taking a risk = to gain a reward, enhance profit, improve market position, customer satisfaction, ...

risk has basic three components: ① probability ②
① event ② probability (varies from project to project) ③ amount of stake / impact

* probability would decrease throughout the project and impact would increase throughout the project.
* do risk management in a regular and continuous basis during your project (iterative process).

Risk identification = brainstorming, WBS (work breakdown structure), procurement management plan, delphi method, SWOT analysis (e.g., organizational strength and weakness, project opportunity & threat, ...), checklist, Ishikawa / fishbone diagram / cause-effect diagram,

Risk categorization: = technical, quality, performance, project management, organizational, external (e.g., rate of inflation, ...),

* everyone looks risk in somewhat differently. (risk overlap)

Assumption and risk = * assumptions are potential risks if your assumptions are incorrect.
* assumption can turn into risks if assumption is wrong.
* do assumption analysis (investigation)

qualitative risk analysis: (high, medium, low)
probability, impact

prioritized list of risks

quantitative risk analysis: (putting numbers) : how much impact is going to have
: looking strategy to respond to those risks

* find some expert (Delphi technique)

* statistical independence:
- mutual exclusivity :- two conditions can not exist at the same time.
$P(A) + P(B) = 1$
- expected (monetary) value ≡ EMV = value associated with risk

risk prone = to take risk, | risk averse = person not willing to take risk

decision tree analysis = expected value diagrammed out into tree
- branches of mutually exclusive events
- components on decision tree
- begins with little box in the left hand side

PMP- (Time management)

① concentration on –
– ✓ Critical path
– ✓ least cost & most gain amount time

② fast tracking: → (do more things concurrently =) looks overlapping tasks

③ → ✓ (concurrent engg. → concurrently we are analyzing the things that we need to do, getting ready to do that works.) = (looks like overlapping whole phases)

simulation
- monte carlo's { Addrisk S/w, Risk plus S/w }
- @ S/w

④ Best tool for planning your project schedule :
shows dependencies, relationships & inter- among activities
✓ (network chart diagram)

⑤ milestone = a particular junction in time (significant in your project)
(zero duration)

⑥ resource allocation & leveling

⑦ resource constrained schedule (fixed resources) → use: resource histogram = Gantchart
time contrained schedule (fixed date)

⑧ simulation (Monte Carlo) [iterative] → planning time

⑨ estimates (duration) ✓ consider no. of resources
– consider level of resources (capability of resources)
– what have been results on similar projects (

⑩ learning curve

⑪ variance =(baseline)

PMP (Project cost management)

① IOR

② Payback period = shortest amount of time that turn around the corner of profit

③ opportunity cost = (in magic number)

④

⑤ work break down structure (WBS)

Risk (methods for S/W systems development)

Risk:
- as a science was born in the sixteenth-century Renaissance, a time of discovery.
- derives from the early Italian word 'risicare' = to dare
- game of chance led to the discovery of the theory of probability, the mathematical heart of risk.
- usually defined as the possibility of loss
- we obtain an instance of risk by specifying values for the risk attributes of probability (the possibility) and consequence (the loss).

Risk Exposure = Probability (likelihood that consequence will occur) X Loss (Consequence, the effect of an unsatisfactory outcome)
(expected value)

Risk management (procedure for resolving risks):
- risk assessment activity (defines the risk) = discovery process of identifying sources of risk, evaluating their potential effects
- risk control (resolves the risk) = process of developing risk resolution plans, monitoring risk states, implementing risk resolution plans, correcting for deviations from the plan.

- Risk management transcends modern management theory, such as Total Quality Management (TQM) and Business Process Reengineering (BPR), because it's basic to decision making.
- Risk management is based on theories that provide different strategies for decision making under probabilistic conditions,
  (Bayes theorem, chaos theory, creativity theory, decision theory, game theory, portfolio theory, probability theory, uncertainty theory, utility theory)

S/W Risk: (a measure of likelihood and loss of an unsatisfactory outcome affecting the S/W project, process, product.)

(1) S/W project risk (operational, organizational, and contractual software development parameters)
- primarily a management responsibility
- includes resource constraints, external interfaces, supplier relationships, or contract restrictions.
- other examples are unresponsive vendors and lack of organizational support.
- Perceived lack of control over project external dependencies makes project risk difficult to manage.
- Funding is the most significant project risk reported in risk assessments.

(2). S/W process risk (includes both management and technical work procedures.

Risk in the large :=

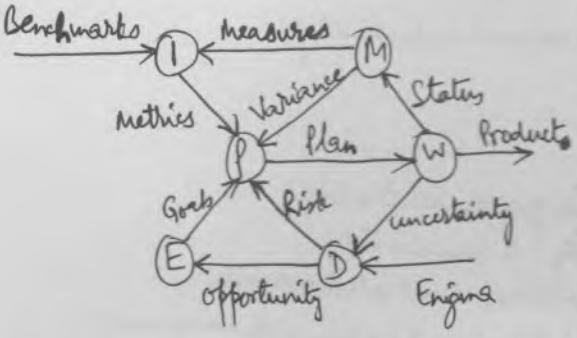
- Risk management must be performed regularly throughout the life cycle of a software system.
  (risks are dynamic i.e. they change over time.)
- ex. (As the project progresses, there is growth in staffing, an increased awareness of project issues, and a different life cycle focus that contribute to the need for routine risk management.)

two perspectives of/struct risk management.

1. risk viewed as extra activity
2. risk viewed as outside activity

routine

1.2.1 The Six-Discipline Model : (6-D) : ⇒

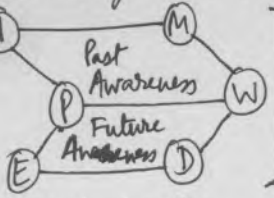


(E) Envision = Develop a vision for a software product.
(P) Plan = Plan the project by mapping resources to the goals established for the software
(W) Work = Produce the product based on the current plan
(M) Measure = Report the variance between expected and actual results to update the plan.
(I) Improve = Analyze benchmarks and organizational project measures to improve processes and metrics.
(D) Discover = assess the uncertainty of our work and external enigma for risk and opportunity, which we manage through changes to the plan and vision.

✓

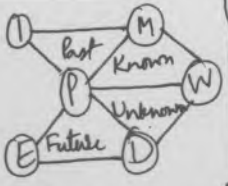
1.2.2 Future Awareness : (reasoning about possibilities) =

— whoever changes the rules wins the game.



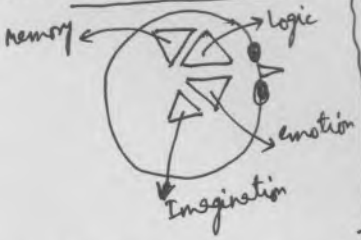
Disciplines for awareness:
Past awareness (PWME) is achieved through planning, working, measuring, and improving. Future awareness (PWDE) is achieved through planning, working, discovering, and envisioning.

1.3 Risk in the Small :



Quadrants of awareness :- The six disciplines form four quadrants of awareness: Known (PWM); Past (PME), Unknown (PWD); and future (PDE)

1.3.1 Personal Awareness : =



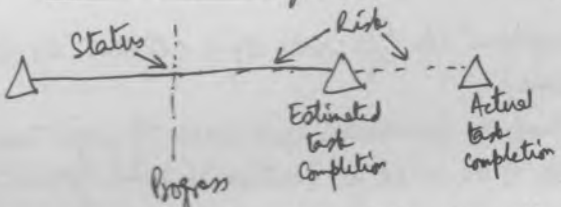
Quadrants of personal awareness.
the six disciplines logically connect four quadrants of our brain.
Logic and memory constitute the left brain, emotion and imagination the right brain.

(Herrmann N. The whole Brain Business Book. New York : McGraw-Hill, 1996)

Logic and memory := factual, logical, rational, theoretical, mathematical, ordered, detailed, sequential, controlled, conservative

Emotions & imagination = musical, spiritual, talkative, emotional, empathetic artistic, holistic, flexible, imaginative, synthesizing

1.3.2 Risk and Personal Progress :- (we make no progress without conquering risk)



Risk and scheduled task completion :- Risk exists in the unknowns of the work remaining to complete a task. Risk also exists in the work plan, because the plan approximates the time and effort required for task completion. This diagram shows how a task estimated to take three months to accomplish may actually take four months to complete without risk management.

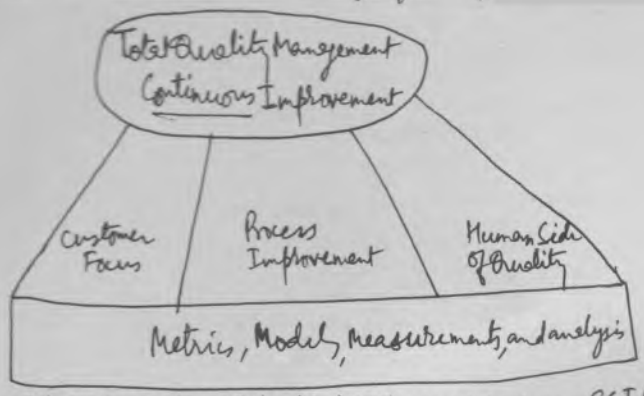
1.4 Consequence of knowledge : (awareness and improved understanding)
- A creative process
- risk management

1.5 Consequences of Ignorance :- lack of skills (to grapple with the risk),
lost opportunity (to perceive it)
suffer from mistakes (do not control your risks)
pain of regret (to learn and practice risk management)

p^2I^2 Success formula :- (maintain focus on the four critical success factors of risk management— people, process, infrastructure, implementation)

Metrics and models in S/W Quality Engineering

TQM:



Lines of Code Defect Rate: Shipped Source Instructions (SSI) (current release) = SSI (previous release) + CSI (new and changed code instructions for current release) − deleted code (usually very small) − changed code

→ improve dev. process and reduce product defects, reduce the non-def. or.

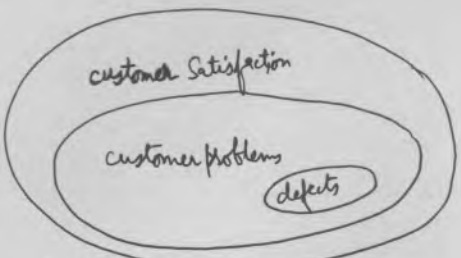
Customer Problem Metric:

Problems per User Month (PUM) = [Total problems that customers reported (true defects and non-defect-oriented problems) for a time period] / [total number of license-months of the software during the period]

(usually calculated for each month after S/W is released to market, and also for monthly averages by year.)

(= number of install licenses of the S/W × Number of months in the calculation period)

: to increase the sale



Scopes of three quality metrics

Defect Removal Effectiveness: $DRE = \frac{\text{Defects removed during a development phase}}{\text{Defects latent in the product}} \times 100\%$

(defects removed during the phase + defects found after delive

Backlog Management Index (BMI) $= \frac{\text{Number of problems closed during the month}}{\text{Number of problem arrivals during the month}} \times 100\%$

Percent delinquent fixes $= \frac{\text{Number of fixes that exceeded the response time criteria by severity level}}{\text{Number of fixes delivered in a specified time}} \times 100\%$

eal-time elinquency Index $= \frac{100 \times \text{Delinquent}}{(\text{Backlog} + \text{Arrivals})}$

Motorola:

Schedule Estimation Accuracy (SEA) = $\frac{\text{Actual project duration}}{\text{Estimated project duration}}$

Effort " " (EEA) = $\frac{\text{Actual project effort}}{\text{Estimated project effort}}$

} Improve project planning

Total Defect Containment Effectiveness (TDCE) = $\frac{\text{number of prerelease defects}}{(\text{number of prerelease defects} + \text{number of postrelease defects})}$

Phase Containment Effectiveness for phase i (PCEi) = $\frac{\text{number of phase i errors}}{(\text{number of phase i errors} + \text{number of phase i defects})}$

} Increase defect containment

Failure Rate (FR) = $\frac{\text{number of failures}}{\text{execution time}}$ ⟹ (increase S/W reliability)

In-Process Faults (IPF) = $\frac{\text{In-process faults caused by incremental software development}}{\text{Assembly-equivalent delta source size}}$

In-Process defects (IPD) = $\frac{\text{In-process defects caused by incremental software development}}{\text{Assembly-equivalent delta source size}}$

Total Released Defects (TRD) total = $\frac{\text{number of released defects}}{\text{Assembly-equivalent total source size}}$

Total Release Defects (TRD) delta = $\frac{\text{number of released defects caused by incremental software development}}{\text{Assembly-equivalent total source size}}$

Customer-Found Defects (CFD) total = $\frac{\text{number of customer-found defects}}{\text{Assembly-equivalent total source size}}$

" " " (CFD) delta = $\frac{\text{number of customer-found defects caused by incremental software development}}{\text{Assembly-equivalent total source size}}$

decrease S/W defect density

New Open Problems (NOP) = total new postrelease problems opened during the month

Total Open Problems (TOP) = total postrelease problems that remain open at the end of the month

Mean Age of open Problems (AOP) = $\frac{(\text{total time postrelease problems remaining open at the end of the month have been open})}{(\text{Number of open postrelease problems remaining open at the end of the month})}$

" " of closed " (ACP) = $\frac{(\text{total number of time postrelease problems closed within the months were open})}{(\text{number of open postrelease problems closed within the month})}$

Improve customer service

Cost of ~~Fixed~~ Fixing Problems (CFP) = (Dollar cost associated with fixing postrelease problems within the month) → Reduce cost of nonconformance

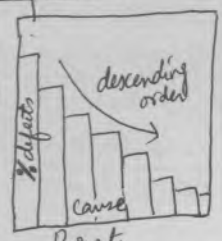
Software Productivity total (SP total) = $\frac{\text{Assembly-equivalent total source size}}{\text{Software development effort}}$

" " delta (SP delta) = $\frac{\text{Assembly-equivalent delta source size}}{\text{Software development effort}}$

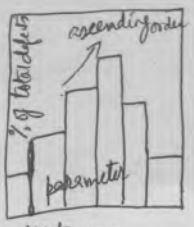
} Increase S/W Productivity

Applying seven basic Quality tools in software Development:

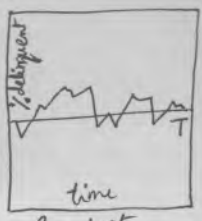
1. Checklist (check sheet)
2. Pareto diagram (20% of causes account for 80% of the defects)
3. Histogram (frequency counts of a sample/parameter)
4. Run chart (parameter states overtime)
5. Scatter diagram (relationship of two interval variables)
6. Control chart (Run chart with control limit/warning limit)



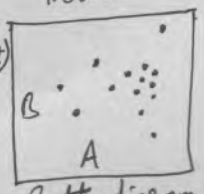
Pareto



Histogram



Run chart



Scatter diagram

Control chart

✓ process capability $C_p = \frac{|USL - LSL|}{6 \text{ sigma}}$

Capability index, $C_p = \frac{|USL - u|}{3 \text{ sigma}} = \frac{|u - LSL|}{3 \text{ sigma}}$

USL = upper engineering specification limits
LSL = lower " "
sigma = standard deviation of the process
6 sigma = overall process variation

✓ control limits for defect rates (u chart):

$$\text{upper limit} = \bar{\mu} + 3\sqrt{\frac{\bar{\mu}}{n_i}}$$
$$\text{lower limit} = \bar{\mu} - 3\sqrt{\frac{\bar{\mu}}{n_i}}$$

$\bar{\mu}$ = value for central line = cumulative defect rate across subgroups (weighted average of defect rates)

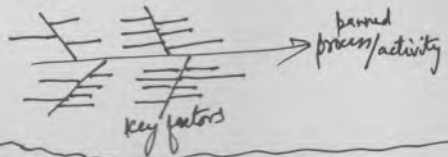
n_i = size of subgroup i for calculation of defect rate (number of lines of source code or number of function points)

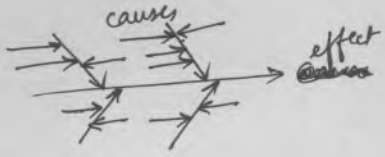
✓ control limits for percentages (e.g., effectiveness metric):

$$\text{upper limit} = \bar{p} + 3\sqrt{\frac{\bar{p}(1-\bar{p})}{n_i}}$$
$$\text{lower limit} = \bar{p} - 3\sqrt{\frac{\bar{p}(1-\bar{p})}{n_i}}$$

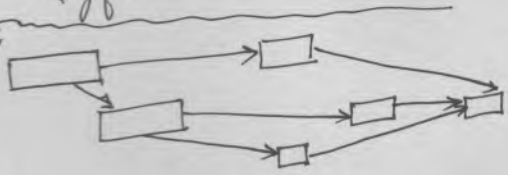
$\bar{p}$ = centre line = weighted average of individual percentages
n_i = size of subgroup i

7. Cause-effect diagram (Fishbone diagram):





8. Relation diagram:



error detection efficiency $= \frac{\text{errors found by inspection}}{\text{total errors in product before inspection}} \times 100\%$

defect removal efficiency $= \frac{\text{defects found by removal operation}}{\text{defects present at removal operation}} \times 100\% = \frac{\text{defect found}}{\text{defect found + defect found later}} \times 100\%$

early detection percentage $= \frac{\text{no. of major inspection errors}}{\text{total no. of errors}} \times 100\%$

E, effectiveness of activity (development phase) $= \frac{N}{N+S} \times 100\%$

N = no. of faults (defects) found by activity (phase)

S = " " " " " by subsequent activities (phases)

TDCE, total defect containment effectiveness $= \frac{\text{no. of prerelease defects}}{\text{no. of prerelease defects + no. of postrelease defects}}$

PCE_i, phase containment effectiveness $= \frac{\text{no. of phase } i \text{ errors}}{\text{no. of phase } i \text{ errors + no. of phase } i \text{ defects}}$

defect removal effectiveness for each development step $= \frac{\text{defects removed (at the step)}}{\text{defects existing on step entry + defects injected during development (of the step)}} \times 100\%$

Q, (number of defects in the released S/W − defects found in the field (customer usage)) $= \frac{\text{Total defects for the life of S/W}}{\left[\frac{\text{(major problems found during review/inspections and unit testing)}}{\text{(Problems tracking report after code integration)}}\right]^2}$

Software engineering-

Software process – set of activities that leads to the production of a software product.
e.g., software specification, software design and implementation, software validation, software evolution, -

Software process models: – waterfall model, evolutionary development, component-based SW engg. (component analysis, requirements modification, system design with reuse, development and integration)

Process iteration: – incremental delivery, spiral development
– spiral development (objective setting, risk assessment and reduction, development and validation, planning)

Process activities: – software specification (requirement engineering) – (feasibility study, requirements elicitation and analysis, requirements specification, requirements validation)
– SW design and implementation (Architectural design, Abstract specification, Interface design, component design, data structure design, algorithm design)
– SW validation (validation and verification) – (component or unit testing, system testing, acceptance testing)
– SW evolution

Rational Unified Process: – RUP perspectives (dynamic with time phases, static process activities, good practices during process)

CASE (Computer-Aided SW engg.) – e.g. (tools like design editors, data dictionaries, compilers, debuggers, system building tools and so on.), (workbenches like IDE, programming, testing), (environments like integrated programming)

* provides automated support for software processes.
– supports individual process activities; workbenches support a set of related activities; environments support all or most software activities.

Project management: –

– project planning (plan (introduction, project organization, risk analysis, hardware and SW resources requirement, work breakdown, project schedule, monitoring and reporting mechanisms)
– project scheduling (bar charts and activity networks)
– risk management (project risks, product risks, business risks)
– these risks types overlap.
– SW risks (staff turnover, management change, hardware unavailability, requirements change, specification delays, size underestimation, CASE tool underperformance, technology change, product competition)

risk management stages/levels:
– risk identification (technology risk, people risk, organizational risk, tool risk, requirement risk, estimation risk)
– risk analysis (likelihood and consequence)
– risk (response) planning (Avoidance strategies, mitigation (minimisation) strategies, contingency plan)
– risk monitoring (constantly assess and plan, revision)

= SW requirements – (user requirements, system requirements)

functional, non-functional, domain requirements

functional requirements – (what the system should do)

non-functional requirements – (define constraints, reliability, response time, store occupancy)
① product requirements = (product behavior)
② organizational requirements = (policies and procedures in customers and developer's organizations)
③ external requirements = (factors external to the system and its development process)

domain requirements – (application domain)

user requirements – (functional and non-functional requirements without detail technical knowledge.)

system requirements – (expanded versions of the user requirements with detail technical knowledge

* system requirements (specification) notations :– structured language specifications, design description languages, graphical notations, mathematical specifications

Interface specification – (interface between existing system and your system)
① ~~proc~~ procedural
② data structures
③ representations of data

* SW requirements document =
① intro. (purpose, scope, definitions, acronyms, abbreviations, references, overview)
② general description (product perspective, functions, user characteristics, general constraints, assumptions and dependencies)
③ specific requirements (functional, non-functional, interface requirements
④ Appendices
⑤ Index

Chapters – Preface, intro, Glossary, user requirements definition, system architecture, system requirements specification, system models, system evolution, Appendices, index

---

Architectural design :-

System organization → ① repository model ② client server model ③ layered model
modular decomposition styles → ① object-oriented decomposition ② function-oriented pipelining
control styles → ① centralized control ② event-driven systems
Reference architectures → ① Generic ② Reference (domain-specific)

Extreme programming (XP) –

practices: incremental planning, small releases, simple design, test-first, refactor, pair programming, collective ownership, continuous integration, sustainable pace, on-site customer

---

Managing people = (consistency, respect, inclusion, honesty)

- selecting staff
- motivating people
- managing groups (group composition, group cohesiveness, group communication, group organization)
  working environments
- People Capability Maturity Model (P-CMM) = 5-level model (initial, repeatable, defined, managed, optimized)
  * provide a framework and associated advice for improving the capabilities of people in an organization and improving the organization's capability to gain benefits from its human assets,

SW Cost Estimation =

= software productivity (size-related metrics, function-related metrics)
= estimation techniques (Algorithmic cost modelling, expert judgement, estimation by analogy, Parkinson's law, pricing to win)

① Algorithmic cost modelling ($Effort = A \times (Size^B \times M)$

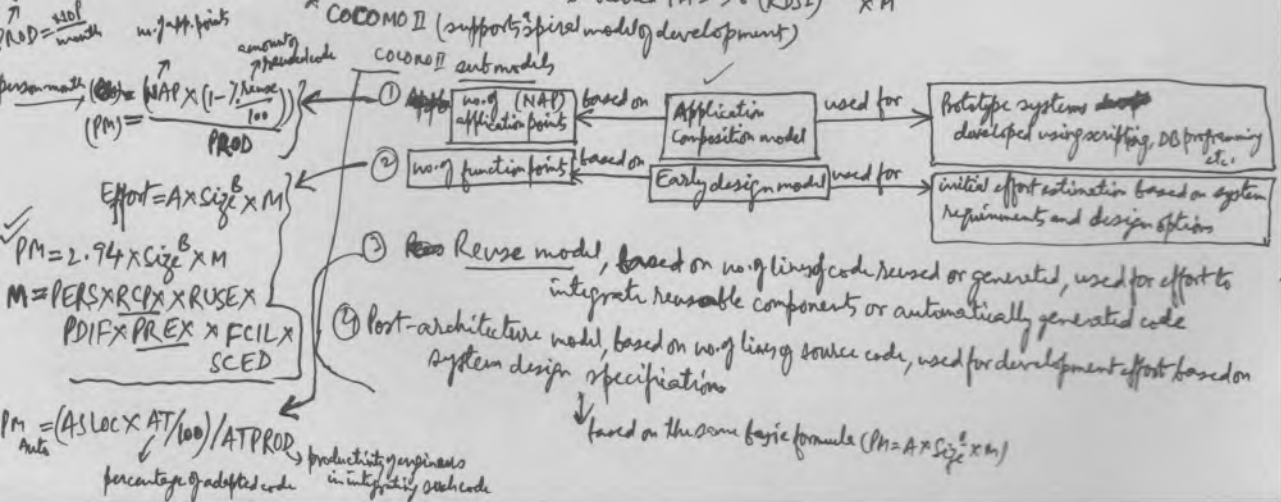
A = constant factor that depends on local organizational practices and the type of SW that is developed.
Size = either an assessment of code size of SW or functionality estimate expressed in function or object points.
B = value of exponent usually lies between 1 and 1.5
M = multiplier made by combining process, product, and development attributes, such as dependability requirements for SW and experience of development team.

no. of thousands of LOC (in KSLOC) ← source

– COCOMO model (empirical model)

* COCOMO I (COCOMO 81) – person-month
  – simple $PM = 2.4\,(KDSI)^{1.05} \times M$
  – moderate $PM = 3.0\,(KDSI)^{1.12} \times M$
  – embedded $PM = 3.6\,(KDSI)^{1.20} \times M$

* COCOMO II (supports spiral model of development)



Project duration and staffing:

TDEV (to estimate calendar time required to complete a project) $= 3 \times (PM)^{(0.33+0.2*(B-1.01))}$

↓ effort

cocomo model:

cocomo II: $TDEV = 3 \times (PM)^{(0.33+0.2*(B-1.01))} \times SCED\ Percentage / 100$

↓ percentage increase or decrease in nominal schedule

QUALITY MANAGEMENT:-

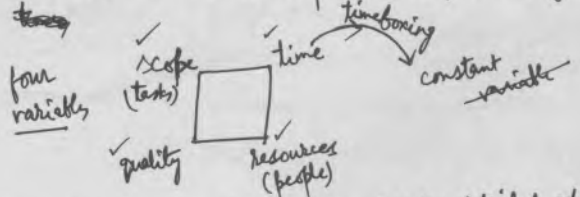
- quality assurance
- quality planning
- quality control

* Process and product quality
* quality assurance and standards

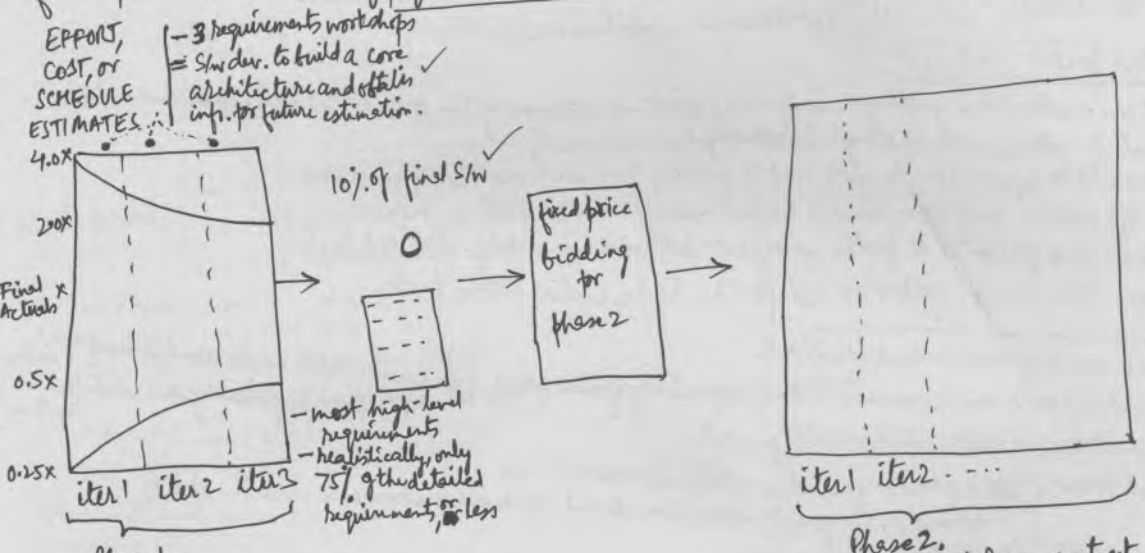
Agile & iterative development

iterative and evolutionary:

- iterative and incremental development (IID)
- length of one iteration (1–6 weeks)
- most projects have 3 iteration
- risk-driven and client-driven iterative planning (choose riskiest element for earlier iterations & client dependent choice of features)
- timeboxed iterative development (except scrum of 30 days, in most of IID method, not all timebox lengths need be equal) e.g.: first iter. maybe four weeks, second be 3 weeks



- evolutionary and adaptive planning (a detailed schedule is not created beyond a relatively short time horizon, so that the level of detail and commitment is commensurate with the quality of information.)
- fixed-price contracts (running project in two contract phases, each of multi-timeboxed iterations) (e.g.: in UP)



- Incremental delivery (the practice of repeatedly delivering a system into production (or the marketplace) in a series of expanding capabilities.)
  - use often between 3 and 12 months.

* (the result of each iteration are not delivered to market place, but the results of an "incremental delivery are.)

* (evolutionary delivery – to capture feedback regarding the installed product, and use this to guide the next delivery.
  - to best meet some difficult-to-predict need, such as the most frequently requested new features.
  - no fixed plan of future deliveries; each is dynamically created based on emerging info.

Specific IED (iterative & evolutionary methods) =

- EVO
- UP (e.g.: RUP) based on risk-driven development in early iterations, core architecture
- MS Solutions Framework from Microsoft
- OPEN
- Win-win Spiral Model and MBASE Spiral Model from Barry Boehm

Agile (classification of methods)

- barely sufficient ceremony (for agile methods) [XP (1-4 weeks), UP (2-6 weeks), Scrum (no), ...]
  (the amount of method weight in terms of documentation, formal steps, reviews, and so forth.)
- Agile manifesto (individuals and interactions, working software, customer collaboration, responding to change)

Agile Project Management:

1. deliver something useful to the client; check what they value.
2. Cultivate committed stakeholders.
3. Employ a leadership collaboration style.
4. Build competent, collaborative teams.
5. Enable team decision making.
6. Use short timeboxed iterations to quickly deliver features.
7. Encourage adaptability.
8. Champion technical excellence.
9. Focus on delivery activities, not process-compliance activities.

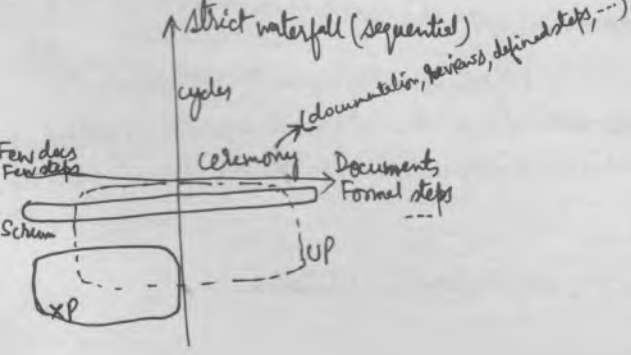
XP-oriented project practices:

1. Guiding Vision - establish a guiding vision for the project and continuously reinforce it through words and actions.
2. Teamwork & collaboration through leadership and community.
3. establish and support the team's set of guiding practices, such as XP or Scrum.
4. Provide visible and open access to project management and other info.
5. Apply just enough control to foster emergent behavior in a self-directed team.
6. Reinforce the vision, follow or adapt the rules, listen to the people.

- embrace communication and feedback
- value individuals and communications (especially face to face), [Scrum's daily meeting and common project room, XP's pair programming and whole team together.]
- do simplest thing that could possibly work.
- empirical process (- high-change and unstable domains.
  - based on frequent measurements and dynamic response to variable events.)
- guided by Agile manifesto and principles.

= Agile team as a Complex Adaptive system (CAS) = self-organizing teams with collective decisions.

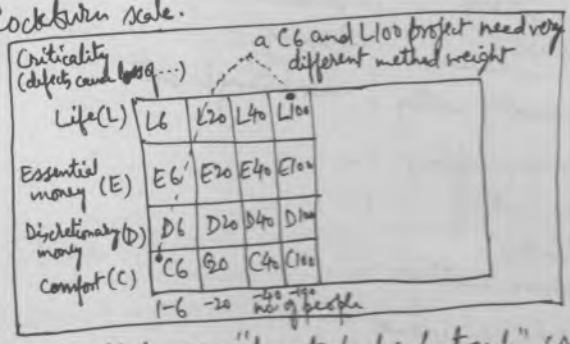
Agile Hype: - adaptable, collaborative, delivery driven, people oriented, customer focused, guided by a vision; develop in short iteration;
XP practices: (embracing changing requirements, communication, self-organizing teams, adaptive planning, ...)
start: (test-driven development and continuous integration)



Specific Agile methods:

- Scrum: (self-organizing teams, daily team measurement, avoidance of following predefined steps (prescription from outside), daily stand-up meeting (Scrum meeting), 30-calendar-days iteration, demo to stakeholders after each iteration)
- XP: (collaboration, quick and early software creation, skillful development practices, communication, simplicity, feedback, courage.)
- Crystal Methods: (peopleware, communication, education, ---)
  (different versions of Crystal (Clear, yellow, ---) contain increasing method weight (or process ceremony in terms of defined and ordered steps, document, reviews, etc.) as a function of staff size, criticality, and project priority.

Cockburn scale.



e.g., E6 means a project of 1-6 people, where the worst that can happen from a system failure is loss of essential money.

Agile modeling: (collaborative "low tech, high-touch" creation of barely good enough, disposable models to aid understanding and communication.)

- Create several models in parallel
- use the simplest tools
- take a picture, erase the board
- update only when it hurts (e.g., developing the code is more important)
- model with others
- display models publicly (e.g., project management data on the walls)

Other methods and practices:

- Adaptive Software Development (ASD)
- Dynamic Solutions Delivery Model (DSDM)
- Feature-driven Development (FDD)
- Lean Development
- Pragmatic Programming

Story: wish list of features → agreed fixed first release date, varying features → first iteration plan (for feedback)

requirements and planning workshop ← 20% most architecturally significant, risky, and valuable items (chosen requirements) ← 20% req. analysis in detail ← functional (use cases), non-functional (clear & measurable goals)

analysis, estimation, enter tasks into Scrum Sprint Backlog spreadsheet

daily meeting, Agile modeling → begin coding → (version control server) check out, unit test, check complete acceptance test, build machine for continuous integration service → daily track the remaining work/effort, update (Scrum Sprint Backlog spreadsheet), cross out completed tasks ×

all tasks finished, demo

---- ← planning ← II req. workshop ← start second iteration

Scrum

self-directed and self-organizing team, daily team measurement, avoidance of prescriptive process, (no external addition of work to an iteration, once chosen), daily standup meeting with special questions, usually 30 calendar day iterations, demo to external stakeholders at end of each iteration, (each iteration, client-driven adaptive planning), daily build (integration & regression test)

lifeCycle : (composed of (four) phases, (4)

Pre-Game		Development ✓ (3)	Release ✓ (4)	
Planning ✓ (1)	Staging ✓ (2)			
establish the vision, set expectations, and secure funding	identify more requirements and prioritize enough for first iteration	- implement a system ready for release in a series of 30-day iterations (Sprints)	- operational deployment	Purpose:
- write vision, budget, initial Product Backlog and estimate items - exploratory design and prototypes	- planning - exploratory design and prototypes	- Sprint planning meeting each iteration, defining the Sprint Backlog and estimates - daily Scrum meetings - Sprint Review (demo)	- documentation - training - marketing & sales - - -	Activities:

Product Backlog :- (all conceivable items go in the backlog and are prioritized by the Product Owner. The estimates (in person-hours of effort) start as rough guidelines, refined once the team commits to an item.)

Req.	Num	Category	Status	Pri	[illegible]
[illegible]	17	feature	underway	5	2
[illegible]	232	use case	not started	5	60
- -	12	issue	complete	4	10
- -	14	defect ...	underway	3	2

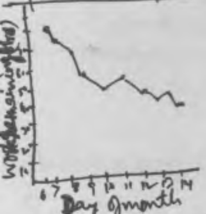
- all items for the product
- * the release backlog is a subset of product backlog.
- includes features, use cases, enhancements, defects, technologies...

Sprint Backlog :- (the daily estimate of work remaining for each task; with date and total hours remaining on each day.)
- updated daily by responsible members or by a daily tracker who visits each member.
- new estimates are allowed to increase above the original estimate.
- tasks for the iteration.
- granularity 4-16 hours.

task description	originator	responsible
meet to discuss	JM	JM/SR
- -	TL	- -
- -	..	- -
- .		-

Status	Hours of work remaining					update
	6	7	8	9	10 [illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	→ hours estimate
Complete	[illegible]	.	.	.		
not started	8	8	8	8		
in progress	[illegible]	[illegible]	8	7		

✓ Sprint Backlog Graph :- visual summary of estimated task hours remaining in the sprint Backlog. ✓
- most critical data to track ✓
- post an updated version of this each day on the wall by the scrum meeting. ✓
- estimate of work remaining (in hrs) versus days.



Management practices :- (Backlog Graph, Sprint Backlog, Product & Release Backlog)

Scrum Master :=
- conducts the sprint review (demo).
- " daily scrum meeting.
- listen to progress and removes impediments.
- mediates between management and scrum Team.
- ensures Scrum values and practices followed.
- knows and reinforces the project and iteration vision and goals.

- sprint planning (iteration plan).
- sprint (usually 30-day).
- blocks (problems) reported at meeting are ideally removed before the next meeting.

* (Scrum + XP) = X Breed

Extreme Programming: (collaboration, quick and early software creation, skillful development practices)
(communication, simplicity, feedback, courage)
- TDD, planning game, small and frequent releases, system metaphors, simple design, testing, frequent refactoring, pair programming, team code ownership, continuous integration, sustainable pace, whole team together, coding standards, daily meeting

Life Cycle:

Exploration	Planning	Iterations to First release	Productionizing	Maintenance
(Enough well-estimated story cards for first release.) (Feasibility ensured.)	- Agree on date and stories for first release.	- implement a tested system ready for release.	- operational deployment	- Enhance, fix. - Build major releases.
- prototypes - exploratory proof of technology programming - story card writing and estimating	- Release planning game - story card writing and estimating	- testing and programming - iteration planning game - task writing and estimating	- documentation - training - marketing - - - .	- may include these phases again, for incremental releases.

Story Cards := (handwritten note on a paper index card) during the planning game, many of them are written.
- minimalist approach to recorded requirements
- record user stories := features, fixes, non-functional requirements (e.g. documentation),
- usually in the one-day to three weeks range of estimated duration.
- usually represent features, not use-cases or scenarios

Find lowest Fare 3

task lists := generate a list of tasks for all stories chosen for the iteration (on whiteboard)
- another popular alternative is to generate individual task cards.
- once a task is chosen, enter an effort estimate (in ideal engineering hours) - tasks should be in the 1-2 day range.

Visible Graphs := - to easily communicate - to the team - something they find useful to measure.
- e.g., acceptance tests defined and passing; story progress, and task progress.

Project management := task list, visible graphs, ~~storycards~~ (on the wall, in order to better communicate. Contents depends on teams, e.g., number of tests defined vs. passing), story cards.

management: [XP Coach := (process conscience, process customizing, intervention, teaching)
[Tracker = (collect metrics, tell progress, feedback on poor estimates)

Customer = [- writes stories and acceptance test (customer test)
[- picks stories for release and for iteration

Tester = [- helps customers write and develop tests
[- develops tests

Programmer := writes tests, design, code, refactors, identifies tasks and estimates

Core practices: [whole team including customer work together] [small and frequent releases] [automated acceptance (functional) tests (all tests (acceptance and unit) must run with 100% pass) (all result of acceptance tests written with collaboration of customer define a testable statement of what acceptance mean (customer tests)] [test-driven / first dev.: unit test written first & before code to be tested (test → code)], [Coding standards: everyone needs to follow the same coding style], [System metaphors: capture overall system or each subsystem with memorable metaphors to describe the key architectural themes.]

other practices := Onsite customer proxies (substitutes), Customer on call (access via mobile phone, ...), embrace change, tasks are volunteered assigned chosen by people, very light modeling, just enough (minimize documentation), metrics (number) completed tasks and stories, number and success rate of running tests), (~30 minutes)

Unified Process (iterative process framework, optional activities, and work products (artifacts) e.g.: vision, risk list)
- short timeboxed iterations, high-risk and high-value elements (e.g., core architecture) in early iterations,
- leverage reuse of existing components, ensure that you deliver value to your customer,
- accommodate change early in the project,
- work together as one team.

↑ optionally, can have absorbing practices from agile (XP, Scrum, ...)

= UP organizes iterations within 4 phases. The elaboration phase iterations emphasize programming the ~~basic~~ risky core architecture, construction phase iterations build the remainder.

* Sample UP disciplines and work products :-

Discipline	work product	Comment
Requirement	Vision	Summary of stakeholders' key need and features.
	Use-case Model	The set of use cases describing the intended functions and environment.
Design	Design Model	An object model describing the hardware and software realization of the use-cases in terms of collaborating objects.
	Software Architecture Document	A system overview or learning aid that includes several architectural views.
Project management	Iteration Plan	The goals and tasks for the current or next iteration.
	Risk List	A list of prioritized risks with associated mitigation plans.

* Sample partial UP development case :=

Discipline	Techniques	Artifact	Incep. I1 (Iteration → I1)	Elab. E1..En	Const. C1..Cn	Trans. T1..Tn
Requirements	one-day timeboxed requirements workshops, prototypes, paper-based UI mockups;	Vision	S=start	R=refine		
		Supplementary specification	S	R		
Design	Pair designing doing whiteboard sketches, captured with camera, test-first design, reverse engineering.	Design Model		S	R	
		SW Architecture Document		S		
Project Mgmt.	All Scrum management practices	Risk list	S	R	R	R
Implementation	Pair programming, test-first development, continuous integration	code, graphics, etc.	S	R	R	R

Life Cycle of UP :=

inception
- purpose: high-level objectives, business case, vision, and scope defined and agreed
  - ~10% of significant requirements defined in detail
  - key risks identification
  - elaboration effort estimation
- milestones: (establishing a common vision) Agreement on scope, vision, priorities, risk identification, a plan to start elaboration exists.
- activities: requirements workshop, start vision and risk list, start use-case model and supplementary spec, prototyping

elaboration
- purpose: core, architecturally significant parts of system coded and tested.
  - significant risks identification and mitigation, 80% of major reqs. evolved & defined in detail, enough stability and information to estimate iteration & effort
- activities: testing, programming, designing in short iterations.
  - requirements workshops, refining the vision, refining the environment (process and technical)

construction
- purpose: system completed and ready to deployment, efficient and predictable development, building on the stable architecture coded in elaboration
- activities: testing, programming, designing in short iterations,
  - stakeholder evaluation and steering; ideally only minor req changes
  - create all documents, alpha testing

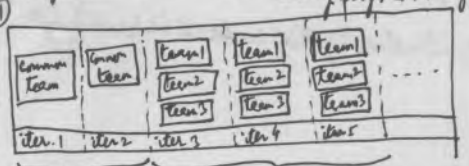
Transition
- purpose: System verified as ready for deployment, deployed system
- activities: beta or release candidate testing and feedback, final programming and documentation, educating & marketing ..., deployment

UP project management := SW dev. plan (overall and composite plan, milestones, resources, etc.)
- iteration plan (the detailed plan for the next iteration, not a plan of all iterations.)
- risk list (risks with severity and mitigations)
- requirement workshops (for requirement)
- early architecture, prefer existing components
- manage requirements

Six best practices :=
- (timeboxed iterations, program high-risk and high-value elements, continuously verify quality by test-driven and continuous integration,
- (visual modeling (sketching and using several engineer, manage requirements, manage change by configuration management and version control ...

Project management tips:- ① multiteam or multisite early development (clarify major components, their collaborations and interfaces through early programming and testing)

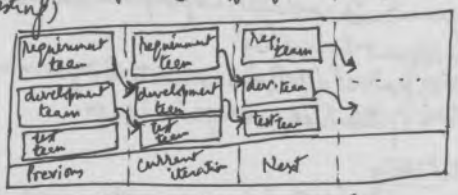
①



small team in common project room, composed of representatives from the sub-teams

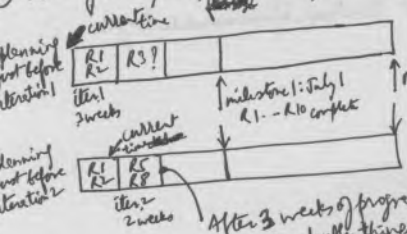
representatives break up, form sub-teams, and lead the remaining work

↑ multiteam development ↑



↑ overlaping or pipelining iteration work ↑

② follow Rolling wave/adaptive planning (there is no detailed plan for all future iterations. usually, we primarily plan in detail for just the next iteration.) flexible



After 3 weeks of programing and a second requirements workshop we suddenly realize, with fresh and better information, that the most valuable things to do in iteration 2 are R5 and R8, in order to most effectively and with lowest risk, move towards milestone 1.

③ planning: consider ending on "wednesday" (or thursday) the goal date for baselining the release. Allow "thursday and friday" as slack days for probable problems or for group planning for next iteration

planning: whole team planning meetings

planning: workers estimate the tasks to finish

planning: wideband Delphi technique: kick off meeting for discussing the source documents and project, units of estimation, get at least three people (or three pairs) to estimate.

2 each person (or pair) creates estimation. They can take minutes or days depending on the scope and can use any method (e.g.: cocomo, micro-estimation, etc...).

3 each person/pair creates three estimates (PERT estimate). 1) most likely 2) optimistic 3) pessimistic

4 (meeting) Facilitator collects then estimates from each estimator and display them with averages. The owners of the estimation are not revealed to the group. Finally, each estimator discusses their insights, problems, and assumptions.

5 Repeat 2, 3, 4 at least once. This is iterative estimation refinement. The point of this step is to provide the feedback to drive adaptation and ~~importance~~ improvement in the next iteration of estimation.

6 Calculate with PERT formula using averages from the final cycle:

$$\text{Estimate} = \frac{\text{Optimistic} + \text{Pessimistic} + 4\,\text{Most Likely}}{6}$$

$$\text{Likely Deviation} = \frac{\text{Pessimistic} - \text{Optimistic}}{6}$$

e.g.: 1

e.g.: 4

	(persondays) Pessimistic	Most likely	Optimistic
— Round 1 ✓ estimates →	(900, 1000, 950)	(800, 900, 850)	(700, 800, 800)
averages →	950	858	767
— Round 2 estimates	(1000, 1100, 1050)	(900, 950, 875)	(800, 900, 850)
averages	1050	908	850

planning: multiple iteration planning meetings, usually near the end of iterations for next iteration and detailed task generation (in the half-day to two-day range), taking photo of each version.

planning: don't forget to budget for iteration overhead tasks like risk management, iteration demos, iteration planning meetings, a group iteration assessment, a daily scrum meeting, some average rate of unexpected defects, infrastructure failures, and so on. All these tasks consume time, and need to be reflected in the iteration task list.

planning: people estimate their time budget each iteration

planning: promote volunteering for tasks rather

planning: show all iteration tasks on a wall or whiteboard.

Heuristics

'iteration goals': risk management, wide coverage of many components development, prefer to functions of high business value, skills development by learning

iteration goals: (what to rank?) use case scenarios, features, defects, and nonfunctional requirements (e.g., internationalization) can all be ranked with the above criteria of risk, coverage, and so on.

iteration goals: How to rank? - Dot Voting

iteration Goals: How to rank? Quantitative methods (numeric values, weights, weighted sums, ...)

iteration Goals: Related iteration length? usually, 1-6 weeks as per small or large teams)

iteration Goals: get technical environment (source control management system, continuous integration build machine, the application server, project Wiki, and so on..) before the first development iteration

iteration Goals: build confidence by easier-first approach and following easier practices in first iteration, ..

iteration goals: divide use cases and scenarios into smaller tasks as fit to complete in different iterations, if use cases are longer.

iteration goals: primary and secondary requests

iteration goals: don't add requests to an iteration

- Tracking iteration progress - and compare actual hours with original hours
- Test-driven development
- Tracking and planning (XPlanner: tool www.xplanner.org)

Earned Value Tracking on IID Projects:-

- EV (Earned value) tracking is a cost and schedule progress measurement method.
- to measure progress in terms of the estimates (or budgets), not only in terms of actuals.

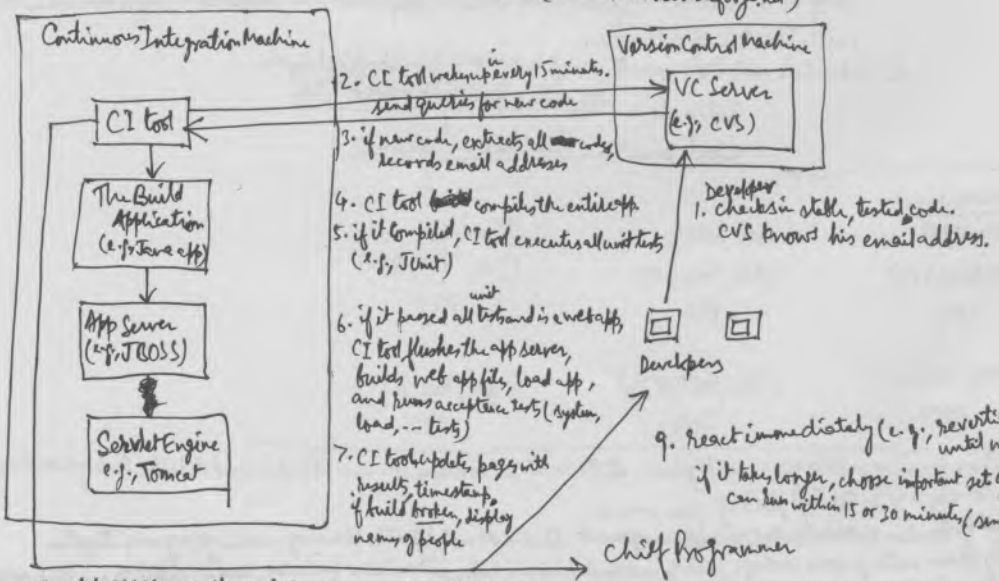
e.g., if creating help web pages for a new system was originally budgeted at 50 person-hours, then when the work is complete, regardless of the actual time spent (for example, 80 hours), the project is said to have "earned" 50 hours of value.

Also, the Budgeted Cost for Work Scheduled (BCWS) - the estimates for future tasks, such as 50 person-hours for the help system.

- Re-calculate the BCWS (estimate) values each iteration, as more information arises. (Rolled baseline).
- simple earned value recognition rule to iteration tasks. (e.g., as soon as an iteration task is underway, it earns 50% of its value)
- Ranking risks and display on boards.
- Managing risks

ENVIRONMENT:-

- Continuous Integration (CI) by tools like, Anthill (www.urbancode.com), CruiseControl (cruisecontrol.sourceforge.net)



- project wiki webs (e.g., www.twiki.org) to publish
- CASE tools and reverse engineering (UML-oriented)
- consider a plotter
- caves and common room (private and common rooms)
- cling sheets / whiteboard paints
- digital cameras, projectors

REQUIREMENTS:-

- Agile modeling
- defining and keeping the vision
- product vision box
- * Moore-style vision statement on the wall/board
  - For (target customer)
  - Who (statement of the need or opportunity)
  - The (product name) is a (product category)
  - That (key benefit, compelling reason to buy)
  - Unlike (primary competitive alternative)
  - Our product (statement of primary differentiation)
- product sheets for marketing and requirements experts, in addition to developers
- evolutionary requirement workshops
- tracking requirements across iterations (e.g., each scenario has a tag such as "3a" or "4c". Thus, track the labels as, "ProcessSale-main success" or "ProcessSale-3a")
- direct user involvement in requirements and products
- Use cases are OK
- quantification is OK (quality requirements is fulfilled..)
- GUIs with Glue [to prototype UIs and UI navigation (in collaboration with clients) using paper, pen, sticky notes, -- on wall]
- Brainstorming (record, not react)
- Brainwriting (write emerging ideas)
- affinity clustering (organize the cards into dynamic sets after brainwriting cards)
- mind maps (hand drawn techniques....)
- team rotation writing (writing ideas on topic by sitting and rotation) → chance discovery = innovator → group

TEST — (test-driven development) = write test (unit test) first and then program codes just enough to make it pass the test)

e.g.

```
public class MoneyTest extends TestCase {
    public void testSimpleAdd() {
        Money m1 = new Money(12, "usd");
        Money m2 = new Money(14, "usd");
        Money expected = new Money(26, "usd");
        Money result = m1.add(m2);
        assertEquals(expected, result);
    }
}
```

- Fit (fit.c2.com) and Fitnesse (www.fitnesse.org) tools for Acceptance testing

1. PERT	(P + 4M + O)/ 6 Pessimistic, Most Likely, Optimistic
2. Standard Deviation	(P - O) / 6
3. Variance	[(P - O)/6]squared
4. Float or Slack	LS-ES and LF-EF
5. Cost Variance	EV - AC
6. Schedule Variance	EV - PV
7. Cost Perf. Index	EV / AC
8. Sched. Perf. Index	EV / PV
9. Est. At Completion (EAC)	BAC / CPI, AC + ETC -- Initial Estimates are flawed AC + BAC - EV -- Future variance are Atypical AC + (BAC - EV) / CPI -- Future Variance would be typical
10. Est. To Complete Percentage complete	EAC - AC EV/ BAC
11. Var. At Completion	BAC - EAC
12. ***To Complete Performance Index*** TCPI	Values for the TCPI index of less then 1.0 is good because it indicates the efficiency to complete is less than planned. How efficient must the project team be to complete the remaining work with the remaining money? (BAC - EV) / (BAC - AC)
13. Net Present Value	Bigger is better (NPV)
14. Present Value PV	FV / (1 + r)^n
15. Internal Rate of Return	Bigger is better (IRR)
16. Benefit Cost Ratio	Bigger is better ((BCR or Benefit / Cost) revenue or payback VS. cost) Or PV or Revenue / PV of Cost
17. Payback Period	Less is better Net Investment / Avg. Annual cash flow.
18. BCWS	PV
19. BCWP	EV
20. ACWP	AC
21. Order of Magnitude Estimate	-25% - +75% (-50 to +100% PMBOK)
22. Budget Estimate	-10% - +25%
23. Definitive Estimate	-5% - +10%
24. Comm. Channels	N(N -1)/2
25. Expected Monetary Value	Probability * Impact
26. Point of Total Assumption (PTA)	((Ceiling Price - Target Price)/buyer's Share Ratio) + Target Cost
Sigma s	• 1s = 68.27% • 2s = 95.45% • 3s = 99.73% • 6s = 99.99985%
Return on Sales (ROS)	Net Income Before Taxes (NEBT) / Total Sales OR Net Income After Taxes (NEAT) / Total Sales
Return on Assets(ROA)	NEBT / Total Assets OR NEAT / Total Assets
Return on Investment (ROI)	NEBT / Total Investment OR NEAT / Total Investment
Working Capital	Current Assets - Current Liabilities
Discounted Cash Flow	Cash Flow X Discount Factor
Contract related formulas	Savings = Target Cost – Actual Cost Bonus = Savings x Percentage Contract Cost = Bonus + Fees Total Cost = Actual Cost + Contract Cost

PMP ITTO (Inputs, Tools and Techniques and Outputs)

- **Integration**:
  - Tools and Techniques are same for all the processes, except Develop Project charter has "Project Selection Methods" and *Monitoring and Control project work* has "Earned value technique".
  - Remember there is no *Expert Judgment* in *Direct and Execute project Work.*
- **Scope**:
  - Scope Definition TT is **APES** (Alternatives Identification, Product Analysis, Expert Judgment and Stakeholder Analysis)
  - Scope Verification has only one TT which is **Inspection**. The only other place where you will find Inspection as TT is in Quality Control process.
- **Time:**
  - Activity Attributes is updated in all the process except in Activity Definition where it is created.
  - Memorize TT for Schedule Development
  - Activity Definition TT are **PERTD (P**lanning component, **E**xpert Judgment, **R**olling wave planning, **T**emplates, **D**ecomposition)
  - Schedule Control TT are **PS PS PV** (**P**rogress reporting **S**chedule Change control System, **P**erformance Measurement, **S**chedule comparison Bar charts, **P**M software **V**ariance analysis)
- **Cost**:
  - Memorize TT For Cost Estimating. Remember that this overlaps with Activity Duration Estimate process.
  - Cost budgeting has Funding Limit Reconciliation as TT which produces output Project funding requirements
  - Cost Management plan is updated in Cost Estimating process
- **Quality:**
  - Quality Planning creates 3 plans in output and 3 other outputs which start with Quality.
  - TT for Quality planning is ABCDC
  - QC has all the outputs of QA + 6 more outputs.
  - There are 10 TT for QC. Out of which there are 7 Tools for QC. Remember Inspection appears as TT here and only other place it appears is in Scope verification
- **HR** - Constraints for HR Planning are **C E O** (Collective bargaining agreements, Economic condition and Organization Structure)
- **Risk**
  - First 4 processes have only one output.
  - Risk Register is updated in all the processes after its created.
  - We recommend that you memorize ITTO for Risk Management process
- **Procurement:**
  - This is a tough chapter and we recommend you keep your focus here and try and **memorize** as much as you can.

General Guidelines:

This is the best way to address ITTO. You should try to memorize it at artifact level, instead of process level. Meaning you should know everything that happens with Project Charter, WBS, Contract and so on. If you remember this then you will be able to answer many questions in exams.

- **Project Charter** is input for only following:

Preliminary **Scope** Statement
Scope Planning
Scope Definition

- **Resource Calendars -** There may be questions in exam asking you about resource calendar. You should remember that it is
  - Input for Activity Duration Estimates, Schedule Development and Cost Budgeting

    - Activity Resource Estimates have Resource calendars as outputs.
- **Project Calendar:** Talking about calendars, remember that project calendar is updated in Schedule Development.
- **Scope Statement -** is input to all the planning processes after its created except
    - Activity Resources Estimating
    - HR Planning
    - Risk Response Planning
    - Plan Contracting

Remember that Scope Statement is **updated** in Integrated Change control process, Create WBS and Scope control

- **WBS and WBS dictionary** are inputs to
    - Scope control
    - Activity Definition
    - Cost Estimating
    - Cost Budgeting
    - Plan Purchases and Acquisitions

Remember that Scope Verification only has WBS Dictionary as input **not** WBS.

- **All Monitoring and Control Processes** Except (Integration Management & Manage Stakeholders) will have following 4 outputs:
- Requested Changes
- Recommended Corrective Actions
- Project Management plan (Updates) (Except in Performance Reporting)
- Organizational Process Assets (Updates)

- **PAW - S**cope control, **C**ost Control, **R**isk Monitoring and control and **C**ontract Administration processes have PAW in Inputs :) . PAW stands for **P**erformance Reporting, **A**pproved change requests and **W**ork Performance Information. The other M & C Processes have two of PAW inputs except (Integration Management & Manage Stakeholders)
    - You can remember this as SCRC has PAW
- **Recommended Preventive Actions**: This occurs in only 4 processes - Monitor and Control Project work, Risk Monitoring and Control, Quality control and Manage project team.
- **Recommended Defect Repair** is output of only two processes
    - M & C project work
    - Quality Control
- **Cost of Quality** - Input to **Q**uality Planning and TT For **C**ost Estimating.
- **Forecasting**
    - **Forecasts** are outputs of M & C project work and Performance Reporting
    - **Forecasted completion** is output of Cost control and input to Performance Reporting
    - **Forecasting** is TT for Cost Control
- **Five outputs** of *Direct and Manage Project Execution* are inputs to *Quality Assurance* namely
    - Implemented Change requests,
    - Implemented Corrective actions
    - Implemented Preventive actions
    - Implemented Defect Repair.
    - Work Performance Information.
- **Contract** which is output of Select Selles process is input to *Contract Administration* and *Cost budgeting*

MEMORY AIDS FOR PMP EXAM

1. The following are common tools and techniques for all the processes in the Project **Integration** Management knowledge area:

- <!--[if !supportLists]-->• <!--[endif]-->Project Management Methodology
- <!--[if !supportLists]-->• <!--[endif]-->Expert Judgment (**except** Direct and Manage Project Execution)
- <!--[if !supportLists]-->• <!--[endif]-->PMIS

2. After the Project management plan is developed, it is an input for all successive processes in the Project Integration Management knowledge area.
3. Activity Attributes Updates is a common output for all the processes in the Project Time Management knowledge area except the first process where activity attributes are created.
4. **Requested Changes** is a common output for all the processes in the Project Cost, Time, Quality & Scope Management knowledge area except the Activity Duration Estimating Process , Quality Planning and Scope Planning
5. Tools and techniques used in Risk Identification:
    a. BIRDS (Information Gathering techniques)
    b. CSI (Diagramming techniques)

B I R D S:

- Brainstorming
- Interviewing
- Root cause identification
- Delphi technique
- SWOT Analysis

C S I:

- Cause and Effect Analysis
- System or Process Flow charts
- Influence Diagrams

Tools and techniques used in Quantitative Risk Analysis:

6.
    a. PIE (Data gathering and representation techniques)
    b. MEDS (Quantitative risk analysis and modeling)

P I E:

Probability Distributions
Interviewing
Expert Judgment

MEDS:

Modeling and Simulation (e.g. Monte Carlo Analysis)
EMV Analysis
Decision tree analysis
Sensitivity Analysis (e.g. Tornado Diagram)

7. Tools and techniques used in Risk Response Planning:

<!--[if !supportLists]-->a. <!--[endif]-->Strategies for Negative Risks / Threats:

ATM:

- Avoid
- Transfer
- Mitigate

<!--[if !supportLists]-->b. <!--[endif]-->Strategies for Positive Risks / Threats:

SEE

- Share
- Enhance
- Exploit

8. Tools and techniques used in Scope Definition:

APES

- Alternatives Identification

- Product Analysis
- Expert Judgment
- Stakeholder Analysis

9. There is only one Tool & Technique for Risk Management Planning: Planning meetings and analysis.
10. When the seller selection is based on PRICE, its often called a tender / Bid / Quotation. When the seller selection is based on technical skills / approach, its called a proposal.
11. Recommended Preventive action is used to bring the project into compliance with the project plan. Recommended corrective action is anything that needs to be done to bring the seller in compliance with the terms of the contract.
12. Prevention is keeping errors out of the process. Inspection is keeping errors out of the hands of the customer.
13. Attribute sampling is whether the result conforms or not. In Variables sampling, the result is rated on a continuous scale that measures the degree of conformity.
14. Common causes (a.k.a. random causes) are normal process variations. Special causes are unusual events.
15. The result is acceptable if it falls within the range specified by the tolerance control limits. The process is in control if it falls within the control limits.
16. Seven basic tools of quality: CCFRHPS

Take a **C**at **C**at-**F**ish **R**uns on **H**orse **P**ony **S**tallion
Cause and Effect Diagrams
Control Charts.
Flow charting
Run Chart
Histogram
Pareto Chart
Scatter Diagram

17. Constraints for human resource planning:
CEO

- Collective bargaining agreements
- Economic conditions
- Organizational Structure

18. Project Statement of Work describes:
BPS

- Business Needed
- Product Scope Description
- Strategic Plan

POINT OF TOTAL ASSUMPTION (PTA)

A price determined by a fixed price plus incentive fee contract (FPIF) above which the seller bears all the loss of a cost overrun. It is also known as the "most pessimistic cost" because it represents the highest point beyond which costs are not expected to rise, given reasonable issues. If costs go beyond the PTA, they are assumed to be due to mismanagement rather than a worst-case set of difficulties. The seller bears all of the cost risk at PTA and beyond. In addition, once the costs on an FPIF contract reach PTA, the maximum amount the buyer will pay is the ceiling price.

Any FPIF contract specifies a target cost, a target profit, a target price, a ceiling price, and one or more share ratios. The PTA is the difference between the ceiling and target prices, divided by the buyer's portion of the share ratio for that price range, plus the target cost.

PTA = ((Ceiling Price - Target Price)/buyer's Share Ratio) + Target Cost

Example:
Target Cost: 60,000
Target profit: 6000
Target Price: 63,000
Ceiling Price: 65,000
Share Ratio: 70% Buyer and 30% seller
PTA = ((65000 - 63000) / 0.7) + 60000 = 62857

For cost reimbursable contract, Point of Total Assumption (also referred to as break point) is calculated by the following formula: PTA = [{(Ceiling Price - (Target cost + Fixed fee))/Benefit Sharing} + Target Cost]
Example: target Cost = 1,000,000; Fixed Fee = 100,000; benefit/cost sharing = 80%/20%; Price ceiling = 1,200,000
PTA = {(1,200,000 - (1,000,000+100,000))/0.80} +1,000,000 = 1,125,000

PMP FORMULAS FOR EV

AC	CPI=EV/AC	EAC=BAC/CPI
EV=PV*%	CV=EV-AC	ETC=EAC-AC
PV	SPI=EV/PV	
BAC	SV=EV-PV	VAR=BAC-ETC