Doncalko Khoda Review Midterm.

Project scheduling.

Project are late because:

- an unrealistic deadline.
- technical difficulties that would ha
- failure to recognize (correct) project is falling behind schedule.
- miscomunication among project.
- predictable and unpredictable of risks [gnored during planning.
- human difficulties.
- changing from custa
- underestimate of effort or job requiren.

Scheduling principles.

- Compartmentalization
  - decompose product and process into manageable nr of functions
  - define distinct tasks for proces framework activities.

Interdependency.

- etablish tasks interrelation ship.
- isolate critical-noncritical tasks

time allocation

- every task has start and completion date
- every task cosiders interdependent tasks into account.

- effort validation
  must ensure that are enough staff to complete tasks. with the time

- defined responsibilities
  people must be assined specific tasks.

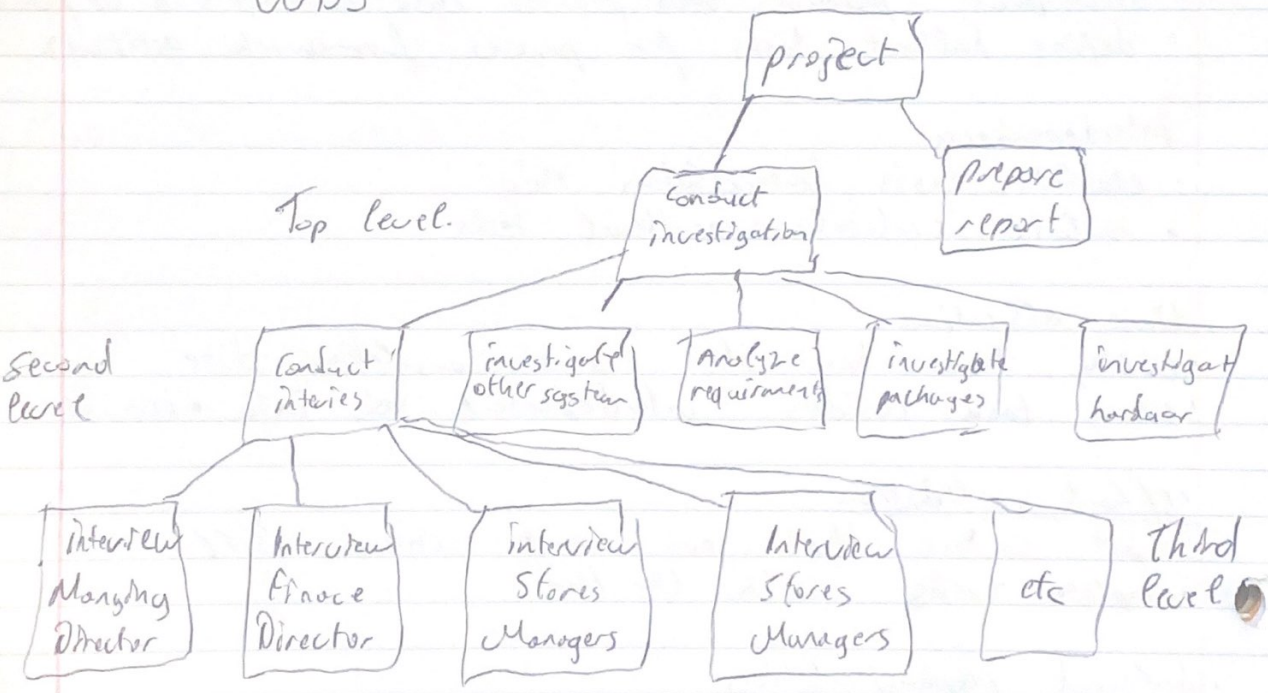
all task should be assigned to people.

- defined outcomes,
  each task must have output.

defined milestones.
all task ~~milestones~~ should be associated with milestone

WBS (work Breakdown Structur.
Provides a framework for organizing and managing approved project scope.
- Helps ensure you have defined all the work that makes up the project.
- Provides a frame work for planing and controlling cost and schedule information.

WBS



Critical Path is the path that take the longest

Reports.

Earned Value Analysis (EVA).

- Is a measure of project development progress.
- enables us to asses the "percent of completness" of a projet using quantitative analysis rather than rely on a gut feeling.
- allows for a accurate reliabl analysis of performance from as early as 15 percent into the project.

BCWS - Budgeted Cost of Work Scheduled

ACWP - Actual Cost of Work Performed.
BCWP - Budgeted cost of Work Performed.

BCWS → Planned cost of the total amount of work scheduled to be performed by the milestone date.
ACWP - Cost incurred to accomplish the work that has been done to date
BCWP - The planned (not actual) cost to complete the work that has been done.
Schedule variance
SV = BCWP - BCWS → negative behind schedule.
cost variance
CV = BCWP - ACWP. → negative is over budge.

SPI : Schedule Performance Index.
SPI = BCWP/BCWS → SPI < 1 → behind schedule

CPI Cost Perfomence Index
CPI = BCWP / ACWP CPI < 1 means project is over budge.

CSI : Cost Schedule Index
CSI = CPI x SPI
The further CSI is from 1.0 the less likely projts recovery becomes.

Cost estimation.
Sofware Project Estimation.
etimation of resources, cost, and schedule for a software engineering effort requires.
• you may not have prior experience and hiltorical information, but you should have a commitment to making a good prediction based on information you have.

Software Cost Estimation
- static approch.
  involves product size.
  is not time dependent.

• dynamic approch
  involves a time factor and may also include product size
-

expert judgment.
domain expert(s) predict based on gut feeling from working on past project

• algorithmic and empirical models.
  equations and algorithms formed after empirical observation
• Process based estimation
• machine learning method
• learn relationship between projects attributes and cost
• based on data collected from past software development projects

Exert Judgment Based.
A = most pesimist
B = most likely estimate
C = most optimists.
E = (A + 4B + C)/6. } weight average.
Problem-based Estimation: LOC/FP Approch.

compute LOC/FP using estimates of information domains values
- use historical data to build estimates for the project.

avg productivity = 620 LOC/pm
Labor = $8000 per month.
cost per line of code 8000/620 = 13.
Total LOC = 33.200
Total estimat project 33 200 · 132 431.000.
Total estimate is 33 200/620 = 54 person-months.

Problem Based with FPA.
function points a mesure of project size in terms of functionality that needs to be implemets.
domain characteristic defined as
- number of extrenal ~~points~~ inputs (EI).
  - user or some inputs.
  - provides distinc applications
  - inquiers are not consider as inputs.
- number of extrenal ~~points~~ outputs (EO).
  - derived with the application
  - provides information to user.
    reports, screens, error most.

domain characteristics

- number of external inquiries (EQ).
  online input that results in the generation of some immediate software response in the form of an online output.
- reperson user interaction. online inputs.
- number of internal logical files (ILF)

number of external interface files (EIF).

$FP = T \cdot (0.65 + 0.01 \times \sum F_i)$

T = unadjusted function point count
F_i = total score from questionnaire.
14 complexity adjustment factor with values 0-5.

Value adjustment factor = $[0.65 + 0.01 \times S(F_i)]$
$= 0.65 + 0.01 \times 57.$
$= 1.17.$

FPA Based Estimation
FP = count-total $\times [0.65 + 0.01 \times S(Fi)] = 64$
55.

avg productivity = 6.5 FP/pm.
labor = 8000 per month.
cost per FP 8000/6.5 = 1230 per-F.
total estimated 1230 x 64 278,720.
total estimated effort 64/6.5 = 10 person-months.

COCOMO
COCOMO I original model.
current COCOMO II current model from COCOMO I

COCOMO I

predicts the efforts and schedule for a software
Three Models. (complexity).
The basic Model.
the intermediate Model.
the Detailed Model.

The Develops Modes : Project Characteristics.
Organic Modes.
stable environment, in familiar.
similar to the previously developed.
relatively small and require little innovation
- Semidetached Mode.
intermediate between Organic and Embedded.
- Embedded Mode.
tight, inflexible constraints, and interface requirements
The product great innovation

Equation
$E = a \cdot S^b * M$.
a and b project type and complexity M is multiplier which depend on various cost drivers.
S is project size in K KDSI or KLOC.
a and b.
↳ project mode: Organic, Semidetached, Embaded.
project type: basic, intermediate, detailed.

The intermediate Model estimate the software develope effort by using fifteen cost driver variables beside the sized variable used in Basic.

Four areas for drivers
Product itself
Computer
Personel
Project itself.

Product Attributes.
RELY → Required Software Reliability.
DATA → Data base size.
CPLX → Software Product complexity.

Computer Attributes
TIME - Execution Time Constraint.
STOR - Main storage Constraint.
VIRT - virtual machine Volatility.
TURN - Computer Turn_Around Time.

Personel Attributes.
ACAP - Analyst Capability
AEXP - Application Experience
PCAP - Programmer Capability.
VEXP - Virtual Machine Experience.
LEXP - Programming language Experience.

Project Attributes
MODP - Modern Programming Practice,
TOOL - Use Software Tools,
SCED - Require Development schedule.

Software Configuration item (SCI)
an approved unit of software code, a document or piece of hardware that is designed for configua management and treated as a destinct eatity in the SCM. process.

Software Configuration item version (SCI version).
the approved state of an SCI at any given point of time during the development or maintenca process.
IEEE Stg 610,12-1990 (specification produce that is approve
Baseline is a milestone in the software development.
- delivery one or more configuration items
- approval of these SCI that is obtained through FTR.

a work product becomes a baseline only after it is revied and approved.

- once the baseline is established each change request must be evaluated and verified before it is processed.

Two issues : what selection of configuratio items
when : when do we start to place entities under configuration control.

Common SCI
- design documents
- software code
  - source code.
  - object code
  - prototype software.
- data file.

- Software development tools.
  compilers and debugger
  application generators.
  case tools.

SCM Process tasks.
1. identification
2. Version control.
3. change control.
4. configuration auditing
5. reporting.

Version Control. four major capabilities.
- project resporitory
- version management capabilities
- make facility.
  issue tracking capobility

Software Configuration Audit.
has the change specified by the ECO been made without modificatio
has an FTR been conducted to asses technical correctness.
was the software process followed and software engineering standat applied.
does the attributes of the configuration object reflect the change.
have the SCM standarts for recording and reporting the change been followed.
were all related SCI properly updated?

CVS (or Subversion) - both open source software
CVS is available as command line or window base
GUI

CVS is version control system import
of SCM, using it you can record the
history of source files and documents.
Ed
git init. directory.

git add .
git commit -m "texts"
git clone.

git pull

git status

Continous Integration to the Rescue.

Before CI
1) The Entire Sorce code was built and then tested
2) Developers have to wait for the test results

3) No Continuous Feedback.

After CI
1. Every commit mad in the source code is built and tested.
2) Developers know the test results every commit made inthe source code. on the run
3) Continous Feedback is present.

Defect tracking: detecting, managing and fixing defects is at the core of improving software quality.

Failure: Any deviation of the observed behavior from the specific behavior

Error: The system is in a state such that further processing by the system can lead to a failure.

Fault: The mechanical or algorithmic cause of an error ("bug").

Validation: Activity of checking for deviation between the observed behavior of a system and its specification

Defect types.

typically classified	specific defect
1. requirements defect.	syntax
2 design defect.	assignment
2 coding defect.	data
3 regression defect.	function
	checking
	interface
	build/package
	documentation
	system
	environment

Defect logging

- describe the situation failure
- plan to work on the issue

work on defect to understand the failure and find the defect.

Defect Management Process (
enter new defect.
review / validate defect
- check to see if this defect is a duplicate.
- set milestone, priority, assign to developer, comment on plan of action
- developer provides estimate of effort needed
- developer marks defect as Open

Defect states.
Pending defect states.
- Unconfirmed: entered by outsider, not yet checked

Open defect states.
New: valid defects, but no work done yet.
Started: developer has started work on the defect
Reopened: the fix failed, work must be redone.

Closed defects states
- Resolved: the developer thinks the has solved the prob
- Verified: SQA has verified that the developer's fix worked
- Closed: all work on this issue has been 'completed, including docs, release notes.

Resolved: developer thinks the solved the prob
- Invalid: The reporter was mistake, user error or misunderstandings.
Duplicate: This defect has already been reported before.
Fixed: the defect was repaired.
Won't Fix: team agrees not to fix the defect.
- later
- Remind.

Usage of Defect Database.

- forcasting development milestone dates.
- Software risk analysis.

Bugzilla is a tracking bug system