

Week-4

L1 Software Proj. Mgmt. - PM Overview

- responsibilities of software manager - POC briefs clients and dev team, form & manages the dev team, proj. scheduling, proj. estimatn, risk mgmt., configura mgmt.

L2 Project Estimation Techniques

- mpt. of estimatn - establish cost & schedule
- key estimatn params -
 1. size of code - KLOC (no. of 1000 lines of code)
 2. effort = person month (effort an indi. can typically put in a month)
- Empirical estimatn techniques (ask people who have completed similar projects)
 - expert judgment (educated guess, estimate cost of diff. components, combine the estimates)
 - duration → human. errors, indi. bias, optimistic, estimates, overlook some factors)
- Delphi technique →
 1. SRS + form for exceeding cost estimate
 2. experts submit their estimates to coordinator
 3. coordinator prepares summary, revise

- Heuristics Tech. (modelled using math. exp.)

G COCOMO (constructive cost estimatn model) - Boehm

$$\text{Effort} = a \times \text{size}^b$$

organic $\rightarrow 2.4 \times (\text{KLOC size})^{1.05}$ person month
 (easy understand proj., small size)

semi-detached $\rightarrow 3.0 \times (\text{KLOC})^{1.12}$ person month
 (mix of exp. + inexp. people)

embedded $\rightarrow 3.6 \times (\text{KLOC})^{1.12}$ pm

(strongly coupled with hardware)

- effect estimator params - people working in proj., technical attributes of proj., tools & pac. of team
→ 15 cost - driver attributes
- tech. is used, depends on seq., type of proj.
- 92% of PMs make estimates based on exp.

L3 Project Scheduling

- import. of schedule - monitor timely completion of task, take corrective acⁿ
- main activities -
 1. identify major activities
 2. break each activity into tasks
 3. determine the dependency among diff. Tasks
 4. estimate for time duration req. to comp. the tasks
 5. represent this info. in charts
 6. determine task starting & ending dates
 7. determine critical path
 8. allocate resources to tasks
- WBS (work breakdown structure) - create tree-like struc., root (proj. name), further children
- Activity Network - diff. activities making up a proj., estimated duratioⁿ, interdependencies, leaf nodes of the WBS become nodes of the activity network
- gantt chart

L4 Risk Mgmt.

- The risk is an anticipated unfav. event or circumstance that can occur while a proj. is underway.
- Syntax errors, 3 party modules, team conflicts

- technical risks - technical aspects, due to dev.
 - (team's insufficient knowledge abt. product
 - ↳ dev. wrong funcⁿs & UI, ⇒ client comm.
 - ↳ shortcoming in exp. comp. ⇒ reg. inspectⁿ
- proj. risks - occurs due to problems in budget, schedule, personnel, resources, & customer-related problems
 - ↳ schedule slippage. ⇒ detailed milestones, itemsⁿ
 - ↳ insufficient know. ⇒ hire proper people
 - ↳ personnel shortfall ⇒ cross training
- business risks - risks which can harm the business aspect.
- PM creates a 'risk table', assigns P (0-1), impact (I) - neg., marginal ... (1-4)

$$\text{Risk} = P \times I$$
, sort (desc)

L5

Project Mgmt. in Agile

- Agile → is divided into items (1-2 weeks), user stories are implemented in each iteration, then prioritized for next itemⁿ, size ⇒ 4-9 people.
- sprint - short, time-boxed period when a scrum team works to comp. a set amt. of work.
- dev. team - req. to comp. work in that given sprint
- prod. owner - interfaces btw. the client & the dev. team
- scrum master - ensure all activities are being done well.

Collaborate

- 2 basic ques.
1. what work can get done in this sprint ?
 2. how will the chosen work get done ?

| M | T | W | T | F | S | S |
|----------|-------|---|---|---|---|---|
| Page No. | YOUVA | | | | | |
| Date | | | | | | |

- sprint planning meeting — 2 hours per week.
- Product backlog - prioritized list of work for the development team that is derived from user stories & cug. How to prioritize? In sprint planning meetings.
- Daily meetings involves all 3 of them. Each member answers 3 questions -
 1. what did I work on yesterday?
 2. what am I working on today?
 3. what issues are blocking me?
- Sprint review → team shows what they have comp., move things to "done"
- Sprint Retrospective → evaluate the last sprint - team dynamics, processes, tools, . User stories that went (didn't go well). Create & implement a plan.
- Key indicators of progress - user stories implemented.
- Project estimation - count the no. of user stories completed per "item" / sprint.
- But, not all user stories eq. same effort. Rate them on a 'scale' 1 → 3 (most complex)
- Velocity - no. of points per "item" / sprint.
- Plan & Document - proj. scheduling early, breakdown into tasks, create milestones.
- Agile - user stories, points, velocity.

L6 Project Tracker Tutorial

- projecttracker.com.