

Project Scheduling

A project scheduler time table that organizes tasks, resources & due dates in a ideal sequence so that a project can be completed on time.

Principles of scheduling

compartimentalization,

interdependency

effort validation

defined responsibilities

defined outcomes

defined milestones.

Performance

EVA

Earned value

is a measure of progress. Percent of completeness provides accurate & reliable readings of performance from 0% early or 100% percent.

Computing Earned value

Bevv → Budgeted cost of work scheduled.

BCWP → Budgeted cost of work performed. (completed)

BAC → Budgeted cost of completion

SPI → Schedule Performance

Index, ratio of earned value to planned value

$ACWP \rightarrow$ Actual cost of work

Performed.

$SV \rightarrow$ Schedule variance

AVF

$CPI \rightarrow$ Cost performance index

Actual cost / planned cost

$EV \rightarrow$ cost variance.

Main term

$$* SPI = BCWP / BCWS$$

Actual progress / planned

$$* SV = BCWP - BCWS$$

difference

$$* CPI = BCWP / ACWP$$

$$* EV = BCWP - ACWP$$

use this to know if project is behind schedule or ahead of schedule

I can find out if value is late or exceeds target

Difference between earned value of actual cost positive if exceed target

Percent scheduled for completion

$SV = BCWS / BAC$ % of work that have been completed by time

Percent complete = $BCWP / BAC$

% of completeness of the project at a given time

WORK

BCWS \Rightarrow indicator measures the progress of the project

sum of budgeted, planned & authorized that have been completed.

BCWP

measures the actual work done

SPI: planned progress are better

actual project is progress

EV

value 1 2500 from

target (check in table, But neg on below 1 2500 behind check/schedule)

CCP-14

project estimation

Main goal ~~to minimize~~ pragmatic strategy for controlling, tracking, monitoring a complex technical project.

covers project to be On time, On budget & quality

Planning test - 1 stories informed

- establish project scope.
- feasibility determine
- Analyze the risks.
- Define required sources.

Determine required Human resources

Define Reusable software resource

Identify environmental resources.

SV \Rightarrow Schedule performance
of the project. Positive value
means project ahead of the

Plan \Rightarrow scheduling plan

and duration estimate \Rightarrow wrong
planning will cause cost overrun
and schedule delay. It must
be done and can't

Planning meeting : E92
located London, UK
Date : 1st June 2000
Budg. 100k, 100 days, 100k

Project planning tools Act-2

Estimate cost & effort

④ Decompose the problem

④ Develop 2 or more estimates, using size function points, process time

④ Resources are estimators

P.E Principles

④ Project scope must be understood.

④ Elaboration is necessary.

④ Historical metrics are very helpful

④ At least 2 different techniques should be used.

④ Uncertainty is inherent in the process.

Conventional methods

① estimation techniques

• task breakdown & effort estimation

• size estimation

• compute LOC/FB using estimates of information domain.

• use option technique to make estimate

for project

Cap-1G

Risk Management

Risk overview

~~its plan~~

~~with goal~~

- Project plans have to be based on assumptions. Risk is the possibility that the assumption is wrong. Then it becomes a problem.
- Risks involve uncertainty & potential losses.
- Risk analysis is intended to help a software development team understand & manage the uncertainty.
- If things go ~~wrong~~ wrong - then make a plan to reduce their impact.

Risk management

Proactive

- Project team reacts to risk when occurs. It mitigates risk to severity of damage.
- mitigation - plan for additional resources to reduce the ~~severity~~ ~~mitigation~~ severity of damage.

Reactive

- fix on failure - Unit testing
- formal risk analysis is performed
- organization connects the source of the risk.

Risk projection using 2m

- Probability
- consequences and system stability

• Project planner, manager, technical staff performing way risk projection

- Probability
- consequences

- Impact
- Accuracy

• Risk subsidies

- Risk tolerance
- Risk threshold

Risk check List

- Product size (\Rightarrow size estimate) significantly ~~more~~ than your goal,
- Business impact \Rightarrow ~~more~~ than 10%
- Customer characteristics \Rightarrow ~~more~~ than 10%
- Process definition \Rightarrow ~~more~~ than 10%
- Development environment \Rightarrow ~~more~~ than 10%
different from the one managed by the customer
①
- Technology to be built \Rightarrow ~~more~~ than 10%
different from standard use; primary new technology
or lack of it
- Staff \Rightarrow ~~more~~ than 10%
size & experience
less than 6 months

Risk management

- 1 • Risk identification
- 2 • Risk analysis & prioritization
- 3 • Risk planning
- 4 • Risk monitoring

①

Use checklist: based on the experience of past projects.

Brainstorming: get knowledgeable stakeholders together.

Causal mapping: identify possible chains of cause & effect.

③

Risk prevention:

Risk Reduction:

Risk transfer

- Risk Reduction Leverage → it is a quantitative means of assessing how risk one being managed.

$$\frac{RE_{\text{before}} - RE_{\text{after}}}{\text{cost of RE}}$$

RE → Risk Exposure

Cap - 9

S.D

Design should exhibit →

Firmness: Should not have any bugs that inhibit.

Commodity: Should be suitable for the purpose.

Delight: experience of using the program should be pleasurable.

S.D is a process to transform user req into some suitable form.

Modularity

It is basically concerned with the attribute of software that allows a program to be intellectually manageable into distinct logical parts. It is technique to divide a software into multiple different & independent modules which can

carry out the task independently.

→ principle: keep complexity away → reduce the complexity along with a lot of effort required to build the software.

User interface: must be simple.

Easy to learn

Easy to use

Easy to understand.

Must edit changes in harmonious with

Design Errors:

lack of consistency (contradict)

too much memorization

no guidance / help

no context sensitivity

obscure / unfriendly

Interface analysis / Understanding

- the people ~~with whom~~ will interact with the system through the interface
- the task that end-users must perform to do their work.
- the content that is presented as part of the interface
- the environment in which this task will be conducted.

(~~and others~~) organized to work with minimum amount of ~~head~~ & something else
definition fast & good or
of business struggle

Interface design principles

Anticipation → web app should design to anticipate the user's next move and offer a smooth transition.

Communication → the interface should communicate the status of any activity initiated by the user.

Consistency → use of navigation controls, menu, icons, aesthetics in each web page to provide a familiar and consistent experience.

Controlled Autonomy:

Efficiency: The design of web app & its interface should optimize the user's work efficiency. Efficiency can be measured by how quickly and effectively the user can perform their tasks.

Focus → web app interface should

focus on the user task at hand

more on navigation

Fitt's Law: - The time to acquire a target is a function of the distance to & size of the target.

navigation to the ERU (Ergonomics)

Human Interface Object: A vast

library of reusable human

interface objects has been developed
of web app

Latency Reduction: - Web app should

use multitasking that let the

work proceed if the operation
is complete.

also provides better user interface

Learnability of web interface

should be design to minimize the learning time then once learned

then minimize the difficulty of it.

Maintain the work product integrity.

WP must be automatically saved

so that it won't get lost if errors occurs.

Readability: All information should be presented through the interface

should be readable.

visible navigation.

navigation should be visible

also navigation should be visible

navigation should be visible

navigation should be visible

Don't make friends of ~~white space~~

344 Assigning a task ^{to} Rutherford style
Emphasize content ^{of} ~~style~~

format ~~style~~ with ^{final} elements from top
Organization ^{and} summary work

left to bottom right -

Biggest task known off ~~bottom~~
Golden Rule

User information controlled from ~~in~~

21 Define interaction time ~~duration~~ in a way that doesn't force user into unnecessary actions

- work on ~~minimizing~~ interruptible tasks
- ~~never~~ provide for flexible interaction.
- don't let interrupting
- Allow user interaction to be ~~allowable~~ ~~done~~ interruptible & undoable.
- ~~initial~~ ~~from~~ ~~old~~ ~~new~~
- Hide technical terms for the casual user.
- Design for the direct interaction with object that appear on the screen.

Cup -11

- Allow user to put the current task into meaningful context.
- Maintained consistency across a family of applications.
- if the past interaction models have exceeded user expectations don't make changes unless there is a compelling reason to do so-

Ques 7(b) Software quality is a defined field of study & practice that describes the desirable attributes of software product.

Software quality attributes are measures for understanding useful features that facilitate the measurement of desired characteristics of a software product.

Ans 8(a) Availability :-

Availability :- it is a measure of planned down time during which system is actually available for use & fully operational.

Ex:- A system shall be at least 99.5% available on weekdays between 6 am to 12 pm.

Performance: It is measured by how well or how rapidly the system must perform specific tasks to fulfill its functions.

Live: Authorization of Atm shall not take more than 10 sec.

Efficiency: It is measured in term of time required to complete any task given to the system.

Like subsystem should utilize proper capacity of disk space & memory efficiently.

etc. etc. etc.

Flexibility: it is measured by ease of adding new capabilities to the product.

Integrity: it comes with security. System integrity should be sufficient to prevent unauthorized access to system functions, prevents information loss, ensures software protected from virus.

Interoperability: it indicates how easily the system can exchange data or services with other system.

It is measured by testing

Reliability: The probability of the software executing without failure for a specific period of time is known as Reliability.

Usability: this can be measured in term of ease of use. The application should be user friendly & it should be easy to learn.

Maintainability

Maintainability: it indicates how easy it is to correct errors or defects or modify the software.

Reusability: Software reuse is a good cost-efficient & time saving development method. Different code library classes should be generic enough to be easily used in different application modules.

Testability: The system should be easy to test & find defects. If required it should be easy to divide into different modules for testing.

Software testing

Testing shows

- Error
- Requirement conformance
- Performance
- An ~~index~~ indication of quality

who test

Developer

- Understand the software
- test gently & ~~driven~~ driven by delivery
- Experiencing the software operation

Independent user

- Must learn about the system
- attempt to break it
- driven by quality
- Exploring the software

✓ & ✓

Validation: its built is traceable to customer requirement.

Verification: software correctly implements a specific function.

Testing Strategy

Code → Unit testing

Design → integration testing

Requirement → validation //

System eng → system //

unit test → जावा कोड ने unit test का किया

करता है एवं Unit जैसे यह test करता

होता है unit testing.

test a small soft unit at a time. performed by the individual

programmer who implemented the unit prior to integration test.

In the integration Test; full design - अनुसार

यह test करता है।

It is a systematic technique for assembling of a software system while conducting test to uncover errors associated with interfacing the module.

Big bang where all units are connected to each other.

TOP - Down \rightarrow Testing starts first time

Test cases are first to be written
and then code is written

Time taken for writing is 42.24
Time taken for writing is 42.24
Time taken for writing is 42.24
Time taken for writing is 42.24

Bottom up \rightarrow Testing time

Test cases are first to be written

Time taken for writing is 42.24

Suppose software is 21 + E

Regression test: Req change \rightarrow Wild changes

Execution of test \rightarrow Subsequent slides
Error findout \rightarrow Test cases
Subsequent slides \rightarrow Test cases

it is re-execution of some subset

of test that have already been

conducted to reinforce the changes

fault cost = 0.0

here too propagated \rightarrow unintended side effect

below are the types of update

• software \rightarrow Conducted \rightarrow some aspect of software configuration changes, \rightarrow configuration changes, \rightarrow configuration changes, \rightarrow configuration changes

then \rightarrow testing \rightarrow ensuring \rightarrow that

that change don't introduce unintended behaviour or additional errors \rightarrow the test cases

if change can be do manually \rightarrow do manually

using automated capture \rightarrow without writing code

Smoke testing

multiple builds \rightarrow multiple builds \rightarrow multiple builds

daily software \rightarrow multiple builds \rightarrow multiple builds

if build is test \rightarrow multiple builds \rightarrow multiple builds

error most likely to happen \rightarrow then next

step to more \rightarrow multiple builds \rightarrow multiple builds

100% \rightarrow 100%

00 - Testing

Unit testing is a linear unit testing

- Operations of ELL are tested
- The state "behaviors" are examined

Integration applies to different

Strategies of test are

- Thread-based testing → respond to input/events
- Use " " " → II interface

- cluster testing → demonstrate I;

spawning collaboration
Higher order testing

System testing → on system integration

Alpha → use by potential user or independent test team.

Beta → Beta test

Accept ৳৮০ ১০৫

Recovery testing

Recovery testing → soft fail

recovery recovery properly performs II verifies

Security testing: proportion mechanism

soft fail built in test

verifies II and generates

System testing: execute in a manner

that demand abnormal quality,

frequency of random volume.

Performance testing: test the run

time performance of software.

invited

organizing Beta testing time

to accomplish the goal

to test soft

Debugging

Some symptom non-corresponding
one or more symptoms for underlying
causes as hidden error.

debugging as symptom to care

এখনো মাত্র \rightarrow PBA এর এর এর

connection \rightarrow সমস্যা হলুব

- Symptom may disappear when another problem is fixed.
- Come from combination of non-error

" system or compiler error

- small assumptions that everyone believes.

④ অন্তে process মানে programme

failure এর কারণ দetermine করা হয়,
then fixed করা হয়।

Techniques

Brute force testing

- most common, least efficient

•

Back tracking

- common, successfully used in many projects.

- code back track \rightarrow কোড করে আগত পদক্ষেপ

Cause elimination

cause hypothesis basis derived

Block box testing

it is a test that only consider the external behaviour of the system. Means here the tester does not know the internal functioning.

Hence the tester attempts to find ~~exist~~ errors like

- i) incorrect or missing function
- ii) interface error
- iii) accessibility error
- iv) behavioral or performance error
- v) initialization & terminal error

The white box testing

it is a method used to test a program of software. In internal functioning, there the tester looks up the code structure of the product to be tested.

- * Here the tester have to ensure that all independent paths within module have been exercised at least once.
- guarantee that all independent paths within module have been exercised at least once.
- Exercise all logical decision on their true & false side.

- Execute all loops at their boundaries.
- Exercise internal data structure to ensure their validity.

এখন independent path exercised অস্থিত
প্রয়োগ করে,

logical decision statement exercise
কাটো ২টা True or false side.

boundaries loop statement execute কাটো

internal data structure statement
কাটো ফাঁকো কাটো