Ridham (75114802719)
5C8

Software Engineering

Assignment - 2

Q1. Can a system ever be completely "decoupled"? That is, can the degree of coupling be reduced so much that there is no coupling between modules?

Sol No, we cannot make a system decoupled. Decoupling means loosening the existing coupling. That is, making sure each component knows as little as possible about the other component around it. Usually, we cannot remove coupling between components ~~completly~~ completely.

Q2. What is the difference between flow chart and a structure chart?

Ans A flow chart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information processing. Flow chart is a convenient technique to represent the flow of ~~contro~~ control in a program.

While a structure chart represents the hierarchical structure of modules. It represents the software architecture that means the various modules making up the system and the dependency. Structure chart representation can be easily implemented using some common programming language. The main focus in the structure chart is on the module structure of the software

Q3. What problems are likely to arise if two modules have high coupling ?

Ans A system ~~whi~~ with high coupling means there are strong interconnections among its modules. If two modules are concerned in high coupling, it means their interdependence will be very high. Any change applied to single module will affect the functionality of other module.

→ Greater the degree of change, greater will be effect on other.
→ As the dependence is higher, any modification will affect modules in a negative manner.
→ Maintainability of the project is decreased.
→ The reusability factor of individual module is decreased, leading to unsophisticated software.

Q4. List the difference of CMM and ISO-9001. Why is it suggested that CMM is better than ISO-9001 ?

Ans ISO only addresses minimum criteria for an acceptable quality system while CMM focuses strictly on software, while ISO 9001 has included hardware, software, processed material & services.

ISO 9001 targets the manufacturing process, although it also includes manufacturing devices but CMM offers a ~~ff~~ model for judging the software processes of an organisation and for identifying key practices required to increase the maturity. This is why CMM is better than ISO-9001.

5. The following parameters for basic & logarithmic Poisson models are given :-

a) Determine the additional failures and additional execution time required to reach the failure intensity objective of 5 failures/CPU hr. for both models.

b) Repeat this for an objective function of 0.5 failure/CPU hr. Assume that we start with the initial failure intensity only.

Basic execution time model	Logarithmic Poisson execution time model
$\lambda_0 = 10$ failure/CPU hr $V_0 = 100$ failures	$\lambda_0 = 30$ failures/CPU hr $\theta = 0.025$/ failure.

Sol Here $V_0 = 100$ failures

$\mu = \cancel{50}\ 5$ failures.

$\lambda_0 = 30$ failures/CPU hr.

Additional failures $= \Delta\mu = \left[\frac{V_0}{\lambda_0}\right](\lambda_p - \lambda_F)$.

a)(i) Basic execution time model

$$\Delta\mu = \frac{V_0}{\lambda_0}(\lambda_p - \lambda_F).$$

$$= \frac{100}{10}(10-5) = 50 \text{ failures}.$$

λ_p (Present failure intensity) in this case is same as λ_0 (initial failure intensity).

Now, $\Delta\tau = \frac{V_0}{\lambda_0} \ln\left(\frac{\lambda_p}{\lambda_F}\right) = \frac{100}{10} \ln\left[\frac{10}{5}\right] = \underline{6.93 \text{ CPU hr}}$

ii) Logarithmic execution time model.

$$\Delta \mu = \frac{1}{\theta} \ln\left(\frac{\lambda_P}{\lambda_F}\right)$$

$$= \frac{1}{0.025} \ln\left[\frac{30}{5}\right] = 71.67 \text{ failures}$$

$$\Delta \tau = \frac{1}{\theta}\left[\frac{1}{\lambda_F} - \frac{1}{\lambda_P}\right]$$

$$= \frac{1}{0.025} \ln\left[\frac{1}{5} - \frac{1}{30}\right]$$

$$= \underline{6.66 \text{ CPU hr}}$$

<u>Logarithmic model has calculated more failure.</u>

b) Failure intensity objective $(\lambda_F) = 0.5$ failure / CPU hr

(i) $\Delta \mu = \frac{V_0}{\lambda_0}(\lambda_P - \lambda_F)$

$$= \frac{100}{10}(10 - 0.5) = 95 \text{ failures}$$

$$\Delta \tau = \frac{V_0}{\lambda_0} \ln\left[\frac{\lambda_P}{\lambda_F}\right]$$

$$= \frac{100}{10} \ln\left[\frac{10}{0.05}\right]$$

$$= \underline{30 \text{ CPU/hr}}$$

ii) Logarithmic execution time model.

$$\Delta \mu = \frac{1}{\theta} \ln\left[\frac{\lambda_P}{\lambda_F}\right]$$

$$= \frac{1}{0.025} \ln\left[\frac{30}{0.5}\right] = 164 \text{ failures}$$

$$\Delta t = \frac{1}{\theta}\left(\frac{1}{\lambda_F} - \frac{1}{\lambda_P}\right)$$

$$= \frac{1}{0.025}\left[\frac{1}{0.5} - \frac{1}{30}\right] = \underline{78.66 \text{ CPU/hr}}$$