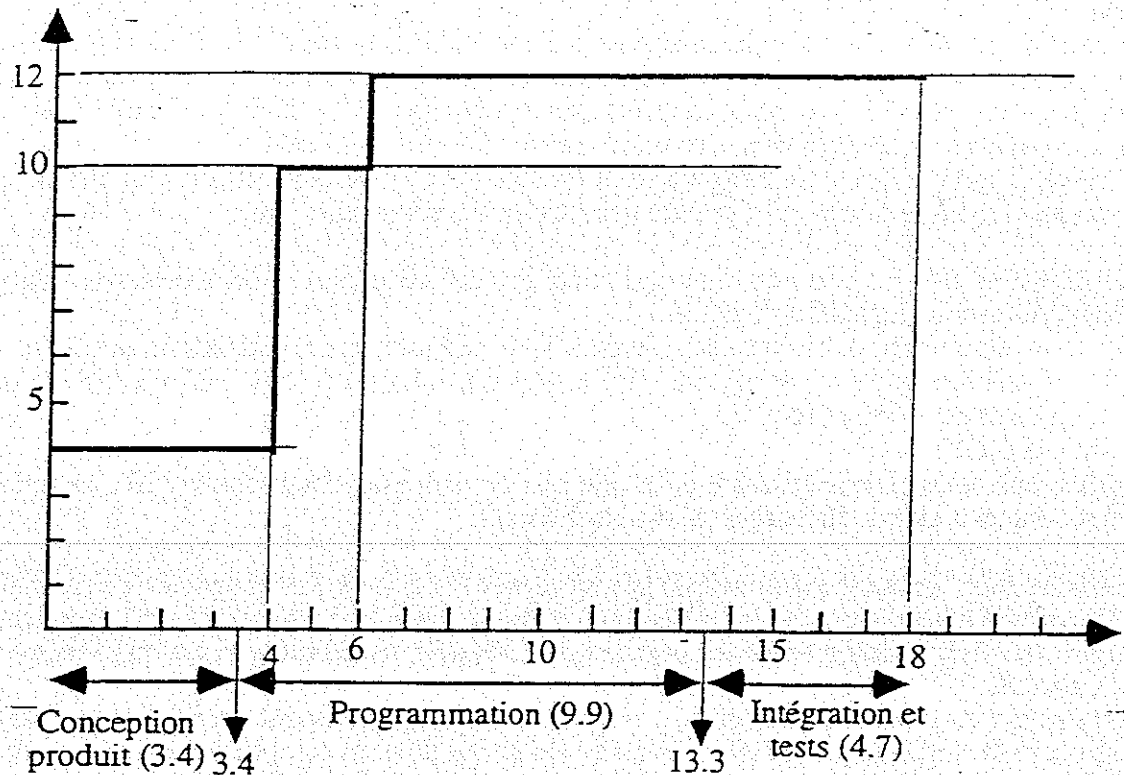


NB : Pour ceux qui ont oublié leur calculatrice $170 / 2.4 = 70.8$ KLS est une assez bonne approximation (C'est un majorant) à 20% près.

Diagramme des effectifs :



QUESTION N° 3 : Identifier les oublis et/ou les impasses de ce chef de projet et de son directeur technique. Quelles sont les conséquences de ces oublis sur le bon déroulement du projet (efforts alloués, ordonnancement des phases, effectifs).

L'oubli, ou l'impasse, le plus notable est la tâche Plan et expression de besoins. La conséquence la plus fâcheuse est que cette activité va être effectuée pendant la phase de conception, ce qui va probablement provoquer des retards. Comme il est prévu de monter à 10 personnes à $T_0 + 4$, il est vraisemblable qu'une bonne partie de l'effort à partir de $T_0 + 4$ va être perdu. Si, par exemple, on considère que la conception est retardée de 1.5 mois et que la moitié de l'effectif ne sait pas exactement ce qu'il doit faire, la perte nette sera de $1.5 \times 5 = 7.5$ HM. Cette perte devra être ajoutée au 9.6 HM des plans qui avaient été oubliés initialement. Il est vraisemblable que ce cafouillage initial sera la cause de nombreuses erreurs qui augmenteront encore la perte.

QUESTION N° 4 : À $T_0 + 9$ mois, le décompte des lignes de code source laisse penser que la taille du logiciel sera de l'ordre de 80 KLS.

Recalculer l'effort et la durée nécessaire au développement de ces 80 KLS. Calculer le surcroît d'effort nécessaire pour terminer le projet.

Le chef de projet décide alors de faire travailler son équipe 10H par jour (au lieu de 8H) et de reporter tous les congés: Est-ce suffisant pour compenser le surcroît d'effort nécessité par le projet ?

À quel moment le chef de projet aurait-il pu, ou du, tirer le signal d'alarme en indiquant à son directeur que ni l'effort, ni les délais de réalisation n'étaient raisonnables ?

Pour 80 KLS, COCOMO nous donne Effort = $2.4 \times (80)^{1.05} = 239$ HM, soit un surcroît de 59 HM par rapport au 180 initialement prévu.

La ventilation par phase donne:

Conception : $239 \times 16\% = 38.2 \text{ HM}$

Programmation : $239 \times 62\% = 148.2 \text{ HM}$

Intégration et test : $239 \times 22\% = 52.6 \text{ HM}$

On voit que l'effort initialement prévu pour la conception est la moitié de ce qu'il aurait fallu.

Le chef de projet s'aperçoit du dérapage à $T_0 + 9$. Il reste donc 9 mois de délai avec un effectif de 12 personnes, soit une production de $9 \times 12 = 108 \text{ HM}$. En travaillant 2H de plus par jour, soit 25% de plus, on peut espérer dégager $108 \times 25\% = 27 \text{ HM}$ de plus, ce qui est notoirement insuffisant; il manque encore $59 - 27 = 32 \text{ HM}$.

Le chef de projet aurait dû réagir dès la fin de la phase de conception, car il était déjà évident à cette date que de gros problèmes allaient rapidement arriver.

QUESTION N° 5 : D'un commun accord, le chef de projet et le directeur technique décide de livrer le logiciel en 2 étapes:

- 1ère version, incomplète, livrée à $T_0 + 18$ mois comme initialement prévu,
- 2ème version, complète, livrée 4 mois plus tard (NB : L'effectif est donc maintenu 4 mois de plus).

Est-ce réaliste ?

Donner une condition nécessaire pour que ce nouveau plan fonctionne (NB : l'ensemble de tous les tests devra être passé au moins 2 fois).

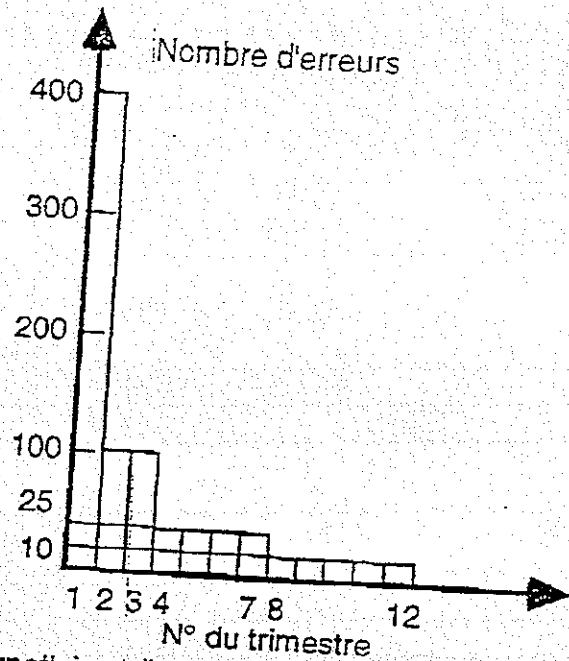
4 mois de plus, en gardant le même effectif remet $4 \times 12 = 48 \text{ HM}$ de plus dans le projet: on a calculé ci-dessus qu'il en fallait 32. Compte tenu des reports de congés, et des nombreuses heures supplémentaires faites par l'équipe, le planning reste très tendu mais est réalisable.

Il est cependant indispensable que les tests aient été soigneusement archivés car il faudra tout repasser en fin de projet. Si ça n'a pas été le cas, le nouveau planning est impossible à tenir.

EXAMEN DE GÉNIE LOGICIEL
Cours B5

Session du 4 février 1997
Locaux : ENSAM

Tous documents et calculatrices sont autorisés.



Si l'on fait l'hypothèse d'une décroissance en progression géométrique, on a pour T2 et T3, respectivement :

$$\frac{N}{x}, \frac{N}{x^2} \text{ soit : } \frac{N}{x} + \frac{N}{x^2} = \frac{N}{2}$$

D'où l'équation :

$$2x + 2 = x^2, \text{ soit : } x^2 - 2x - 2 = 0$$

$$\text{D'où la racine : } x' = \frac{2 + \sqrt{12}}{2} = 1 + \sqrt{3} = 2.732$$

Tous calculs faits, on a donc : $T2 = 400/2.732 = 146$

$$T3 = 146/2.732 = 54$$

Si on applique la même "raison" à la période T4 - T7, on obtient l'équation du 4ème degré :

$$\frac{54}{x} + \frac{54}{x^2} + \frac{54}{x^3} + \frac{54}{x^4} = 100 ; \text{ soit : } \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4} = 1.85$$

On cherche une solution approximative.

Si la progression continue à l'identique, on peut essayer 2.732, soit :

$$\sum_1^4 \frac{1}{x^n} = 0,567 \text{ ce qui est trop faible.}$$

$$\text{Avec } 1,39 \text{ on obtient : } \sum_1^4 \frac{1}{x^n} = 1.877$$

La solution pour T4 - T7 est donc $x = 1,39$

La décroissance de la "raison" de la progression géométrique s'accroît avec le temps, ce qui est conforme aux données du problème (ainsi d'ailleurs qu'à l'expérience).

Question N°2 (3 points) :

Sur la base du modèle COCOMO, reconstituer un profil de charge et de durée vraisemblable (le logiciel est supposé de type P ; pour la répartition par phase vous prendrez "Very Large" dans la table 6-8 ; vous ferez l'hypothèse que pour la programmation et les tests unitaires, la répartition entre programmation et tests est 50%-50%).

Pour 500KLS, les équations du modèle nous donnent :

$$HM = 3.0(500)^{1.12} = 3162HM$$

$$TDev = 2.5(3162)^{0.35} = 42Mois$$

Phase	%	Effort		
Conception Générale	17%	538HM		
Conception Détaillée	23%	727HM		1186HM
Programmation	29%	917HM →	459HM	
+ Tests unitaires			458HM	
Intégration	31%	980HM		1438HM
TOTAL :		3162HM		

Durée :

Phase	%	Durée selon cycle COCOMO
Conception Générale	28%	12 Mois
Programmation	40%	17 Mois
Intégration	32%	13 Mois
TOTAL :		42 Mois

Question N°3 (2 points) :

On considère que pour 1 erreur découverte en exploitation, 20 ont été découvertes en développement : calculer le rendement de l'effort de test (Cf. Leçon Assurance qualité).

Nombre d'erreurs détectées en développement : $20 \times 750 = 15.000$

$$\text{Rendement de la détection} : \frac{\text{Nbre d'erreurs détectées}}{\text{Nbre d'erreurs total}} = \frac{15.000}{15.000 + 750} = 0.95$$

Question N°4 (2 points):

Selon vous, comment peut-on répartir les erreurs entre les 2 processus qui les créent ? On pourra faire l'hypothèse que pour une ligne de spécification, on a 50 lignes de programmation. On considérera que le coefficient d'amplification des erreurs entre les 2 phases est 10 et que le nombre d'erreurs est proportionnel au nombre de KLS.

On peut faire l'hypothèse, comme il a été souvent dit en cours, que le nombre d'erreurs contenues dans un texte est proportionnelle à la longueur du texte, en première approximation.

Pour une programmation de 500KLS, on aura 10KLS de spécification.

On peut donc raisonnablement considérer qu'il y a $15.000/50 = 300$ erreurs de conception qu'il faudra détecter, au plus tard dans les phase de tests.

Si l'on applique le coefficient d'amplification 10, ces 300 erreurs auront générées 3.000 erreurs de programmation.

Le nombre d'erreurs introduite par la programmation est donc : $15.000 - 3.000 = 12.000$ erreurs.

Question N°5 (2 points):

En vous servant du profil de répartition des temps de traitement par défaut du tableau ci-joint, quel est l'effort nécessaire à la correction (On prendra 1 Jour = 8Heures ; 1 Mois = 21 jours) ? Est-ce vraisemblable avec les résultats du modèle COCOMO.

Formuler des hypothèses sur l'effort de création de tests, sur l'effort de dépouillement des résultats et sur l'effort nécessaire à la non régression.

On peut dresser le tableau suivant :

%	Nbre Err	Coût heures	Coût jour	Côut HM
25	3.750	7.500	938	45
50	7.500	37.500	4.688	223
20	3.000	30.000	3.750	179
4	600	12.000	1.500	72
1	150	7.500	938	45
	15.000			564

Le calcul effectué en Q2 donne 1438HM, ce qui est tout à fait compatible avec les 564HM qui ont servi à corriger uniquement les erreurs. On peut considérer que les 874HM restant ont été consacrés à l'écriture des tests et aux tâches logistiques (Non régression, dépouillement des résultats, etc...).

Questions facultatives donnant droit à un bonus (+ 5 points) :

Question N°6 (1 points) :

Comment peut-on organiser les recouvrements des différents processus (Cf. Table 7-1, 7-2 et 7-3 du modèle COCOMO).

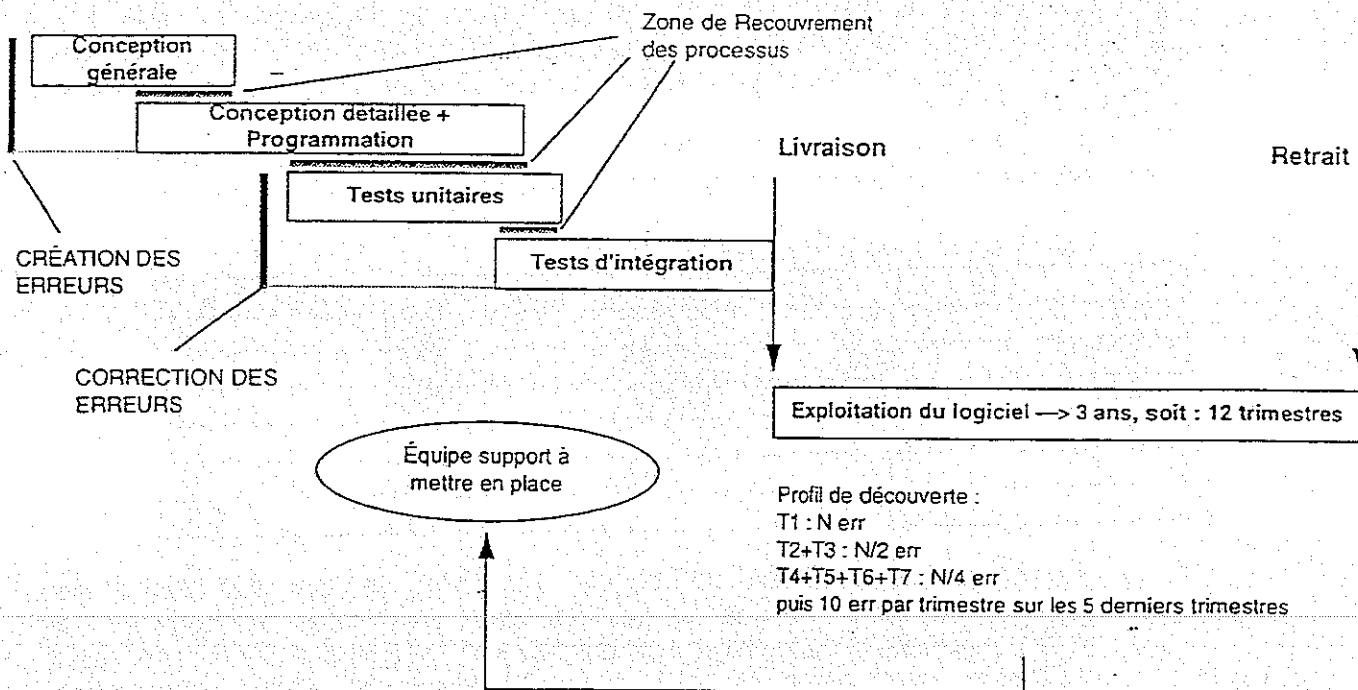
Question N°7 (2 points) :

Quel est le profil de l'équipe de support qu'il faut mettre en place pour assurer une gestion satisfaisante des erreurs découvertes en exploitation et des relivraisons qui en résultent.

Question N°8 (2 points) :

Que se passe t-il, selon vous, si le logiciel est mis en exploitation progressivement sur 100.000 sites (NB : la même erreur peut être détectée sur de nombreux sites ; c'est le phénomène des doublons) ? Que devient l'équipe support : de combien faut-il augmenter les effectifs ?

profil des processus considérés



RÉPARTITION DES TEMPS DE TRAITEMENT PAR DÉFAUT (1/2)

%	Effort moyen par défaut (en heures)	Effort cumulé (en heures)
25%	2h	50h
50%	5h	250h
20%	10h	200h
4%	20h	80h
1%	50h	50h
100 erreurs	6.3h/err	630h

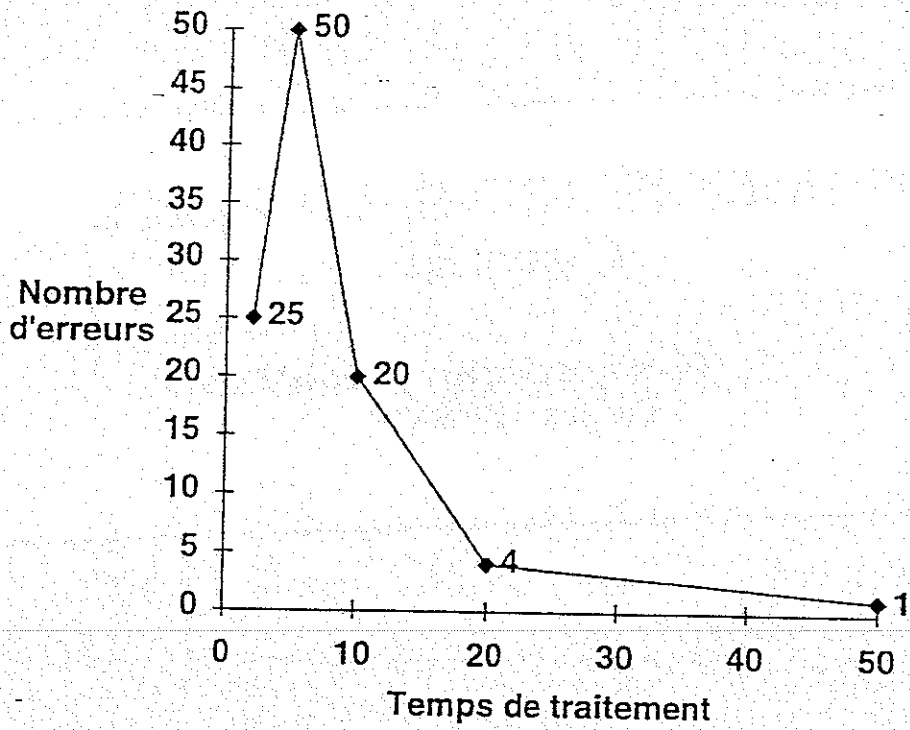
1er Quartile : 8%

2ème-3ème Quartile : 40%

4ème Quartile : 52%

Source : Hewlett-Packard

RELATION DES TEMPS DE TRAITEMENT PAR DÉFAUT (2/2)



EXAMEN DE GÉNIE LOGICIEL
Cours B5

Deuxième Session 11 mars 1997
Locaux : ENSAM

CORRECTION DE L'EXAMEN - 2ème Session

Question de cours (12 points) :

Étude préalable d'un modèle d'estimation du coût des erreurs dans un logiciel de type réseau

Données :

L'équipe de développement réseaux de la société Logiciel Micro travaille en moyenne avec un taux d'erreur de 2.0 Err/KLS.

Elle s'apprête à développer, en langage C, un logiciel client-serveur à partir de protocole RPC¹; la taille de ce logiciel est estimée à 250KLS.

Pour ce développement, on ne considère que les 5 processus suivant :

- Conception générale
- Conception détaillée + programmation
- Tests unitaires
- Tests d'intégration
- Exploitation (sur une durée de 3 ans)

(Voir schéma joint).

Question N°1 (3 points) :

Calculer le nombre d'erreurs découvertes en exploitation par trimestre. Dessiner l'histogramme des erreurs. Sur la base du graphique donner des valeurs vraisemblables pour T2, T3 et T4, ainsi que pour T5, T6, T7, T8, T9.

Nombre d'erreurs : $250 \times 2 = 500$ Erreurs résiduelles.

Bilan des erreurs par plage de temps, soit :

$$N + \frac{N}{2} + \frac{N}{4} + 15 = 500$$

$$N + \frac{N}{2} + \frac{N}{4} = 485$$

$$4N + 2N + N = 1940$$

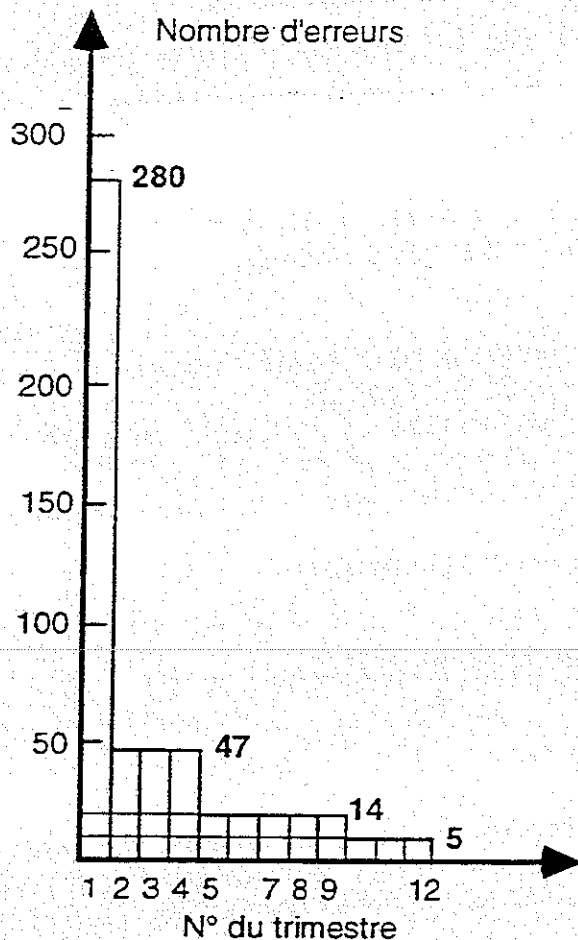
$$N = 277.1 \approx 280$$

On a donc :

T1	280 erreurs
T2+T3+T4	140 erreurs
T5+T6+T7+T8+T9	70 erreurs

D'où l'histogramme simplifié :

¹RPC = Remote Procedure Call (Appel de Procédure Distant). Ce type de protocole permet d'appeler depuis un poste client, des fonctions serveurs comme si elles étaient sur le client.



Si l'on fait l'hypothèse d'une décroissance en progression géométrique, on a pour T2, T3 et T4, respectivement :

$$\frac{N}{x} \cdot \frac{N}{x^2} \cdot \frac{N}{x^3} \text{ soit : } \frac{N}{x} + \frac{N}{x^2} + \frac{N}{x^3} = \frac{N}{2} \text{ ou encore } 280y + 280y^2 + 280y^3 = 140 \text{ avec } y = \frac{1}{x}$$

D'où l'équation du 3ème degré :

$$2x^3 + 2x + 2 = x^3, \text{ soit : } x^3 - 2x^2 - 2x - 2 = 0$$

dont la racine réelle vaut : 2.92

Tous calculs faits, on a donc : T2 = 280/2.92 = 96, T3 = 280/2.92² = 33 et T4 = 280/2.92³ = 11

Pour des raisons évidentes, T4 est obligatoirement > 14 ; il est donc nécessaire de rectifier la distribution.

Si on applique le même raisonnement à la période T5 - T9, on obtient l'équation du 4ème degré :

$$m + \frac{n}{x} + \frac{n}{x^2} + \frac{n}{x^3} + \frac{n}{x^4} = 70 ; \text{ soit } 4 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4} = \frac{70}{n} \text{ avec } n = \text{valeur acceptable pour T5.}$$

On peut procéder par extrapolation linéaire en considérant que la somme est = 70, la moyenne = 14 et que le dernier terme doit être ≥ 5.

Soit :

$$T5 = 23$$

$$T6 = 17$$

$$T7 = 14$$

$$T8 = 10$$

$$T9 = 6$$

On doit donc avoir $T4 \geq 23$, soit par exemple :

$$T2 = 82$$

$$T3 = 33$$

$$T4 = 25$$

L'idéal est bien sur de lisser complètement la courbe.

Question N°2 (3 points) :

Expliquez pourquoi ce logiciel doit être classé dans la catégorie E. Sur la base du modèle COCOMO, reconstituer un profil de charge et de durée vraisemblable (pour la répartition par phase vous prendrez une moyenne 50%-50% entre "Large" et "Very Large" dans la table 6-8 ; vous ferez l'hypothèse que pour la programmation et les tests unitaires, la répartition entre programmation et tests est 50%-50%). Donner un profil de charge vraisemblable pour chacune des phases considérées.

Comme il est dit dans l'énoncé, ce logiciel met en oeuvre une architecture client-serveur, ce qui va avoir pour effet direct de rendre le logiciel tributaire du comportement du réseau. C'est une des caractéristiques du mode E. Comme il a été dit en cours, ce genre de logiciel, en particulier dans sa partie réseau, est beaucoup plus difficile à vérifier et à valider.

Pour 250 KLS les équations du modèle donne :

$$HM = 3.6 \times 250^{1.2} = 2715hm$$

$$TDEV = 2.5 \times 2715^{0.32} = 32mois$$

Pour la répartition, il faut extrapoler entre 128 et 512 (Table 6-8), soit :

Phase	%	Effort		
Conception Générale	18%	489HM		
Conception Détaillée	24%	652HM		991HM
Programmation	25%	678HM →	339HM	1235HM
+ Tests unitaires			339HM	
Intégration	33%	896HM		
TOTAL :		2715HM		

Durée :

Phase	%	Durée selon cycle COCOMO
Conception Générale	38%	12 Mois
Programmation	32%	10 Mois
Intégration	30%	10 Mois
TOTAL :		32 Mois

Question N°3 (1 points) :

On considère que pour 1 erreur découverte en exploitation, 20 ont été découvertes en développement : calculer le nombre d'erreurs découvertes pendant les phases de développement .

Le nombre d'erreurs découvertes en développement est :
 $20 \times 500 = 10.000$

Question N°4 (2 points):

Selon vous, comment peut-on répartir les erreurs entre les 2 processus qui les créent ? On pourra faire l'hypothèse que pour une ligne de spécification de protocoles, on a 100 lignes de programmation C. On considérera que le coefficient d'amplification des erreurs entre les 2 phases est 20 et que le nombre d'erreurs est proportionnel au nombre de KLS.

Si x est le nombre d'erreurs de conception et y le nombre d'erreurs de programmation, on a :

$$20x + y = 10.000$$

Le volume de la conception est $250\text{KLS}/100 = 2.5\text{KLS}$

1ère variante approximative (et naïve) :

Le volume de la conception est $250\text{KLS}/100 = 2.5\text{KLS}$

Si on considère qu'il y a toujours 2 Err/KLS, on a donc 5 erreurs de conception, soit : $20 \times 5 + y = 10.000$ d'où : $y = 9.900$

2ème variante (exacte) :

Si l'on a 2 Err/KLS en fin d'intégration et que pour 1 erreur découverte en exploitation, 20 ont été découvertes en développement, alors le taux d'erreurs en développement est de $2 \times 20 = 40$ Err/KLS. Comme les erreurs de conception ne sont découvertes que pendant la phase d'intégration, c'est donc ce taux qu'il faut considérer et non pas 2 Err/KLS, soit 100 erreurs de conception (au lieu de 5) .

Ce qui conduit à :

$$20 \times 100 + y = 10.000, \text{ soit } y = 8.000 .$$

Soit : 8.000 erreurs de programmation.

Question N°5 (3 points):

En vous servant du profil de répartition des temps de traitement par défaut du tableau ci-joint, quel est l'effort nécessaire à la correction (On prendra 1 Jour = 8 Heures ; 1 Mois = 21 jours) ? Est-ce vraisemblable avec les résultats du modèle COCOMO.

Formuler des hypothèses sur l'effort de fabrication des tests, sur l'effort de dépouillement des résultats, et sur l'effort nécessaire à la non régression : justifier votre répartition. En prenant un coût moyen de fabrication de tests à 10H-Jours, combien y a-t-il de tests ?

On peut dresser le tableau suivant :

%	Nbre Err	Coût heure	Coût jour	Côut HM
25	2.500	5.000	625	30
50	5.000	25.000	3125	149
20	2.000	20.000	2500	119
4	400	8.000	1.000	48
1	100	5.000	625	30

Le calcul effectué en Q2 donne 1235HM, ce qui est tout à fait compatible avec les 376HM (soit = 30%) qui ont servi à corriger uniquement les erreurs. On peut considérer que les 859HM restant ont été consacrés principalement à l'écriture des tests et aux tâches logistiques (Non régression, dépouillement des résultats, mise en place des scénarios de tests avec le réseau, etc...).

Si l'équipe de développement est convenablement outillée, on peut considérer que l'essentiel de l'effort est consacré à l'écriture de tests. En prenant une distribution de Pareto 80-20, on a donc $0.8 \times 859 = 687$ HM pour la production de tests.

Soit un nombre de tests = $\frac{687 \times 21}{10} = 1.443$ tests.

Rapporté au 250 KLS, on a donc $\frac{250.000}{1443} = 173$ LS par test ce qui peut paraître faible.

Questions facultatives donnant droit à un bonus (+ 5 points) :

Question N°6 (3 points) :

La phase d'intégration exige le développement de scénarios de tests qui mettent en oeuvre au minimum un réseau de 2 machines (voir schéma ci-dessous), ou plus. Selon vous quel est le coût de fabrication et de dépouillement de tels scénarios : justifier votre réponse en vous servant des résultats de la question 5.

Quelle méthode de tests vous paraît la mieux adaptée ?

Les scénarios de tests comportent 2 parties : une partie serveur et une partie client. Ils sont donc très difficiles à élaborer. Pour la partie protocole, la méthode des couvertures vue en cours n'est pas vraiment adaptée. Il faut mettre en oeuvre un mécanisme de trace tant du côté serveur que du côté client de façon à s'assurer que les échanges se font correctement. Ces traces doivent être synchronisées, ce qui nécessite une estampille temporelle universelle (même heure sur les 2 machines !).

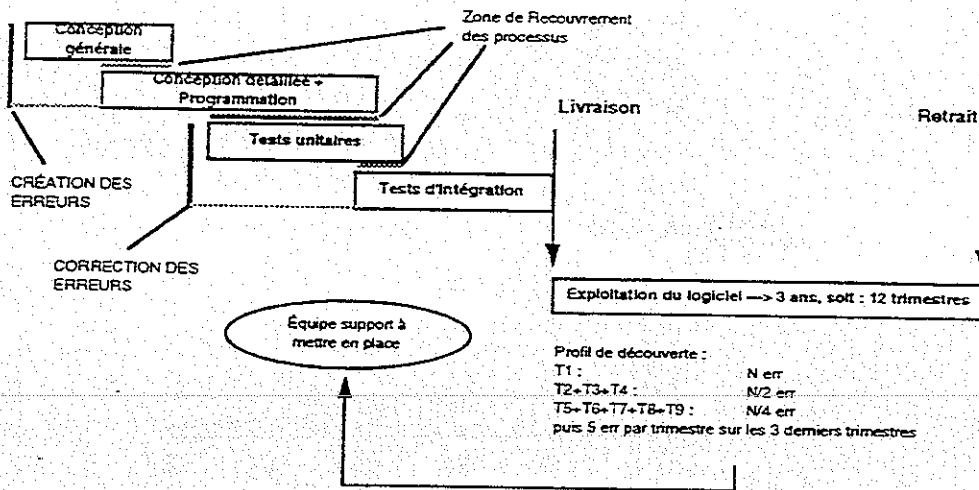
Le coût de dépouillement des résultats des scénarios de tests est donc très coûteux car il faut vérifier la synchronisation à la main. De plus, pour que la non régression soit efficace, il faut archiver une partie de la trace afin de pouvoir procéder à des comparaisons automatiques. Ces comparaisons sont relativement complexes car il faut comparer des plages de valeurs pour tenir compte des événements aléatoires qui peuvent décaler ± fortement la durée des traitements.

Question N°7 (2 points) :

Les 250 KLS du logiciel sont en fait un assemblage de code E et P, dans une proportion relative de 1/3 - 2/3. Comment, et de combien faut-il réajuster le profil du projet que vous avez calculé en 2.

Il faut effectuer une moyenne pondérée en prenant comme poids 1/3 - 2/3, comme cela a été expliqué en cours.

Profil des processus considérés



RÉPARTITION DES TEMPS DE TRAITEMENT PAR DÉFAUT (1/2)

%	Effort moyen par défaut (en heures)	Effort cumulé (en heures)
25%	2h	50h
50%	5h	250h
20%	10h	200h
4%	20h	80h
1%	50h	50h
100 erreurs	6.3h/err	630h

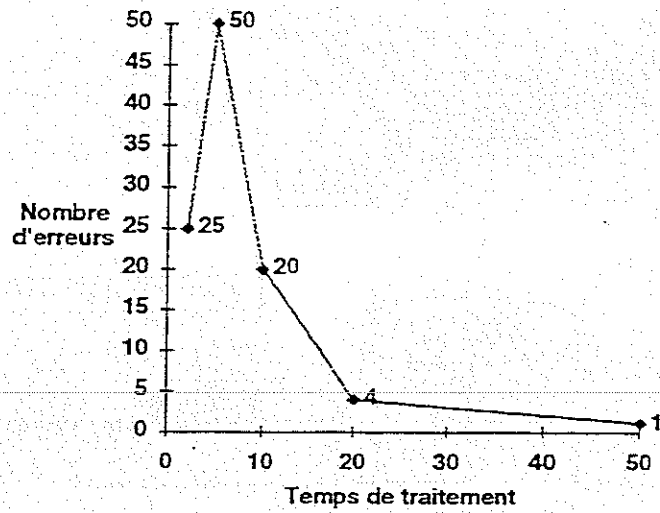
1er Quartile : 8%

2ème-3ème Quartile : 40%

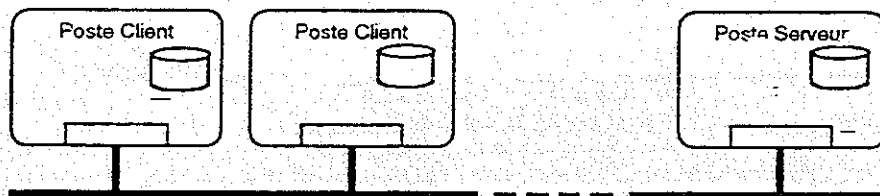
4ème Quartile : 52%

Source : Hewlett-Packard

RÉPARTITION DES TEMPS DE TRAITEMENT PAR DÉFAUT (2/2)



Environnement d'intégration



EXAMEN DE GÉNIE LOGICIEL

Cours B5

1ère Session 28 février 1998
Locaux : maison des examens à Arcueil

La question de cours et l'exercice doivent être traités sur
deux copies distinctes.

Barème de notation :

Question de cours 12 points

Question 1 : 4

Question 2 : 2

Question 3 : 2

Question 4 : 2

Question 5 : 2

Bonus pour les questions 6, 7, 8 : 1 point par question traitée.

Exercice noté sur 8 points

Prenez le temps de bien lire l'énoncé.

Tous documents et calculatrices sont autorisés.

Cours B5 - Génie Logiciel

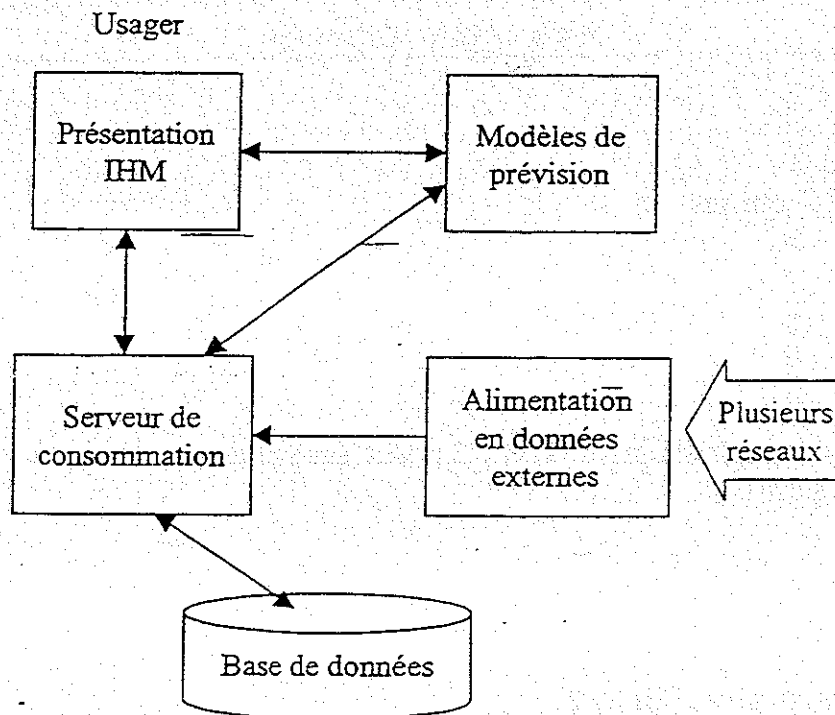
Question de cours

La société FranceElec a réalisé un prototype d'une application destinée à effectuer des prévisions de consommation électrique dont les données sont les suivantes :

	Serveur de consommation	Alimentation en données externes	Scripts (en langage de commandes)	IHM	Modèles de calcul (en Fortran)	Total des lignes source intégrées
Lignes source avec commentaires	17.500	19.400	4.300	84.700	28.000	151.900
Lignes source sans commentaires	10.200	11.800	2.400	54.700	7.000	86.100
Effort en hm	18	10	2	30	Non connu	60

Le code des modèles de calcul est réutilisé sans aucune modification. Le prototype a été réalisé selon une méthode incrémentale telle que celle de la méthode RAD (*Rapid Application Development*). La base de données est une base objets.

L'architecture de ce prototype est conforme au schéma suivant :



Après examen par la direction générale, il a été décidé de transformer ce prototype en application industrielle destinée à équiper 20 centres de prévision qui auront chacun en moyenne 10 copies de l'application, soit 200 systèmes installés en fin de déploiement.