

2 Les tests statiques

- Introduction
- Vie du document
- Tests du documents
- Exemples de codes

Les tests statiques

- Lecture de la documentation
- Lecture et analyse des spécifications
- Analyse statique du code source
- Graphe de contrôle pour déceler les instructions inaccessibles

2 Les tests statiques

- Introduction
- Vie du document
- Tests du documents
- Exemples de codes

Cycle de vie du document

- Spécification
- Conception (qui va faire quoi et avec quels outils)
- Réalisation
- Contrôle
- Archivage, diffusion, exploitation

Cycle de vie du contrôle

- Définition des objectifs et stratégies les mieux adaptées
- Conception
- Réalisation :
 - Réception (composant à contrôler)
 - Préparation (convocations, diffusion des composants)
 - Exercice de contrôle
 - Correction éventuelle (analyse des impacts des défauts, reprise partielle du processus)
 - Évaluation
 - Ajouts éventuels

La lecture du code

- Nécessite une bonne connaissance du langage et de l'environnement
- Il faut connaître les erreurs fréquemment rencontrées pour les détecter plus facilement

Organisation du contrôle de documents

Exemple de lectures

- Le document à vérifier est confié à une ou à quelques personnes pour qu'elles relèvent les erreurs manifestes.
- Ces personnes peuvent faire partie de l'équipe de développement ou non.
- Il n'y a pas de réunion, le document est simplement retourné avec les remarques du lecteur.
- Les erreurs peuvent concerner le fond ou la forme.
- Le résultat de la lecture est pris en compte par le rédacteur du document qui apprécie s'il y a lieu de procéder à des modifications.
- C'est le strict minimum.

La lecture structurée ou revues

Organisation

- Assez de participants pour qu'il y ait beaucoup d'idées, mais pas trop pour que tout le monde se sente concerné
- Distribuer préalablement le produit pour étude
- Adopter des règles et un président
- Se concentrer sur le produit, non sur la personne
- Limiter la durée de la réunion (renforce le cadre formel)
- Ne pas résoudre les problèmes, seulement les identifier
- Rédiger un rapport en ce sens

La lecture structurée ou revues

Bénéfices

- Amélioration de la qualité
- Diminution des efforts
- Échéances mieux respectées
- Amélioration de l'expertise
- Meilleur moral

Organisation du contrôle de documents

Exemple de revue : lecture *walkthrough*

- L'auteur expose le document à un groupe de personne
- Permet de vérifier le caractère exhaustif d'une solution ou d'imaginer une autre solution
- Permet d'intégrer de nouveaux arrivants
- Attention à ne pas réduire la documentation aux transparents présentés lors de ces réunions

Inspections

Organisation

Les inspections ressemblent à des revues mais sont plus formelles, par exemple :

- Utilisation de check-lists pour ne rien oublier
- Les erreurs sont classées par types et enregistrées sur formulaire
- Des statistiques sont réalisées

Inspections

Exemple d'organisation

Quatre membres :

- Le modérateur qui coordonne les activités
- Le concepteur qui a spécifié le document (représente la phase amont)
- Le rédacteur qui a rédigé le document (représente la phase actuelle)
- Le valideur qui joue le rôle de quelqu'un qui utilisera ou mettra à l'épreuve le document (représente la phase suivante)

Inspections

Exemple d'organisation

Six membres (IBM, 1976) :

- Le modérateur qui coordonne les activités
- Le secrétaire ou scribe qui prend les notes et rédige les comptes rendus.
- Les inspecteurs qui évaluent le produit et participent à la rédaction du rapport.
- Le lecteur ou rapporteur qui guide les inspecteurs en rédigeant une synthèse du produit.
- Le vérificateur coordonne la rédaction technique du rapport. Veille à ce que les modifications demandées aient bien été apportées.
- L'auteur apporte sa connaissance du produit.

Conclusion

Bénéfices

- Les techniques statiques n'exécutent pas le code : elles peuvent donc être utilisées à toutes les phases du cycle de vie du logiciel
- De nombreuses études ont montrées que le revues réduisent de façon importante le nombre d'erreurs entre 60 et 95 %

2 Les tests statiques

- Introduction
- Vie du document
- Tests du documents
- Exemples de codes

(Mauvais) exemples

```
int i;main(){for(;i["<i;++i){--i;}"];read('---',i+++ "hell\
o, world!\n",'/'/'/''));}read(j,i,p){write(j/p+p,i---j,i/i);}
```

Code illisible.

Utilisation de spécificités du langage non connues par la plupart des développeurs.

(Mauvais) exemples

```
void analyse(const char * chaine) {  
    char * copie;  
  
    /* On copie la chaîne envoyée en  
       paramètre afin de pouvoir la modifier  
       */  
    strcpy(copie, chaine);  
  
    /* On travaille maintenant sur copie */  
    ...  
}
```

Erreur sur un pointeur.

(Mauvais) exemples

```
#include <time.h>

void date() {
    time_t * t;

    time(t);
    printf("Nombre de secondes depuis l'EPOCH: %d\n", *t);
}
```

Erreur sur un pointeur.

(Mauvais) exemples

```
void test(int x, int y)
{
    if (x>0)
        instruction1;
    else if (y>0 && x>1)
        instruction2;
    else
        instruction3;
}
```

Instruction jamais exécutée.

(Mauvais) exemples

Recherche d'une valeur dans une liste,
on renvoie un pointeur sur le premier élément de la liste
correspondant à la valeur.

```
trouve = false; p = premier;  
while (!trouve)  
    if (p->valeur == v) trouve = true  
    else p = p->suivant;  
return p;
```

Spécification incomplète.

Exemple de spécification : le TDA *Pile*

Les opérations

Générateurs

pile-vide : $\rightarrow Pile$

empile : $T \times Pile \rightarrow Pile$

dépile : $T \rightarrow Pile$

Observateurs

sommet : $Pile \rightarrow T$

est-pile-vide : $Pile \rightarrow \text{Booléen}$

hauteur : $Pile \rightarrow \mathbb{N}$

Exemple de spécification : le TDA *Pile*

Les axiomes

$est\text{-}pile\text{-}vide(pile\text{-}vide) = VRAI$

$est\text{-}pile\text{-}vide(empile(x, p)) = FAUX$

$dépile(empile(x, p)) = p$

$sommet(empile(x, p)) = x$

$hauteur(pile\text{-}vide) = 0$

$hauteur(empile(x, p)) = 1 + hauteur(p)$

Exemple de contrat

Contrat pour la méthode *push(e)* des piles bornées

Précondition :

$$\text{!full}()$$

Postcondition :

$$\text{size}() == \text{oldsize}() + 1$$
$$\text{top}() == e$$

Postcondition incomplète.