Fondamentaux du développeur

Séance 5 - Introduction au débogage - Identification, analyse et correction des erreurs

Philippe ROUSSILLE

Section 1

Objectifs du débogage

Objectifs du débogage

- Identifier rapidement la source d'un problème dans un programme.
- Analyser les causes sous-jacentes pour éviter que l'erreur ne se reproduise.
- Implémenter une correction efficace tout en limitant les impacts sur d'autres parties du code.

Section 2

Étapes d'une méthodologie efficace

Étapes d'une méthodologie efficace

1. **Comprendre le problème** : Collectez autant d'informations que possible sur l'erreur (logs, messages d'erreur, contexte d'exécution).
2. **Reproduire le bug** : Assurez-vous que l'erreur est reproductible de manière fiable.
3. **Isoler la source du problème** : Réduisez l'étendue du code analysé pour localiser précisément l'erreur.

Comprendre l'erreur

- Analysez les messages d'erreur générés par le programme ou l'environnement de développement.
- Posez-vous des questions : l'erreur survient-elle toujours dans les mêmes conditions ?
- Consultez la documentation des bibliothèques ou outils impliqués pour vérifier si leur usage est correct.

Reproduire l'erreur

- Établissez un scénario précis pour provoquer systématiquement l'erreur.
- Identifiez les facteurs externes ou internes qui influencent son apparition (entrées utilisateur, configuration, ressources système).
- Un bug non reproductible est beaucoup plus difficile à corriger ; essayez de réduire l'incertitude.

Isoler la source

- Simplifiez le code en supprimant les parties non essentielles pour observer si le problème persiste.
- Utilisez des breakpoints et l'exécution pas-à-pas pour suivre l'évolution du programme jusqu'à l'erreur.
- Une fois le problème isolé, concentrez-vous sur cette zone spécifique pour comprendre le comportement inattendu.

Section 3

Analyse des causes

Analyse des causes

- **Entrées incorrectes** : données fournies au programme non conformes.
- **Logique défaillante** : conditions mal conçues ou étapes omises.
- **Interactions externes** : fichiers, réseau, ou API tierces.
- Identifiez les hypothèses incorrectes qui ont mené au bug.

Section 4

Correction efficace

Correction efficace

- Appliquez une solution qui traite la cause profonde, pas seulement les symptômes.
- Vérifiez l'impact de la correction sur d'autres parties du code (effets de bord).
- Écrivez des tests automatisés pour garantir que l'erreur ne réapparaîtra pas dans le futur.

Section 5

Erreurs courantes en programmation

Erreurs courantes en programmation

1. **NullPointerException (ou équivalent)** : Une référence non initialisée est utilisée.
2. **Off-by-one errors** : Boucles qui parcourent un élément de trop ou de moins.
3. **Conditions non exhaustives** : Un cas particulier n'est pas traité, provoquant des comportements inattendus.

Stratégies pour traiter les erreurs courantes

- **NullPointerException** : Validez toujours les références avant de les utiliser ; utilisez des annotations ou outils de validation statique.
- **Off-by-one errors** : Revérifiez les bornes des boucles (<= vs <) et ajoutez des assertions pour valider les indices.
- **Conditions non exhaustives** : Ajoutez des clauses par défaut (`else`) ou vérifiez explicitement que tous les cas sont couverts.

Prévenir les erreurs futures

- Adoptez une programmation défensive en validant les entrées utilisateur et en vérifiant les préconditions.
- Utilisez des tests unitaires pour valider chaque fonction de manière indépendante.
- Maintenez une documentation claire et précise pour éviter les malentendus sur les fonctionnalités du code.

Section 6

Conclusion : Développer une approche systématique

Conclusion : Développer une approche systématique

- Un bon débogage repose sur une méthodologie rigoureuse qui combine observation, analyse et tests.
- En identifiant les causes profondes des erreurs, vous améliorez la qualité globale de votre code.
- Prévenir les erreurs dès la conception est tout aussi important que savoir les corriger efficacement.

Merci pour votre attention !

- Des questions ?
- Des points pas clairs ?
- N'hésitez pas.