

Cours Génie Logiciel – M1 Informatique

Partie II – Test logiciel

Chapitre 3 – Techniques statiques

Structure du chapitre 3

3.1 Techniques statiques et processus de test (K2)

3.2 Processus de revue (K2)

3.2.1 Phases d'une revue formelle (K1)

3.2.2 Rôles et responsabilités (K1)

3.2.3 Types de revues (K2)

3.2.4 Facteurs de succès des revues (K2)

3.3 Analyse statique avec des outils (K2)

Test statique (ou revue)

≠

Test dynamique (ou test actif)

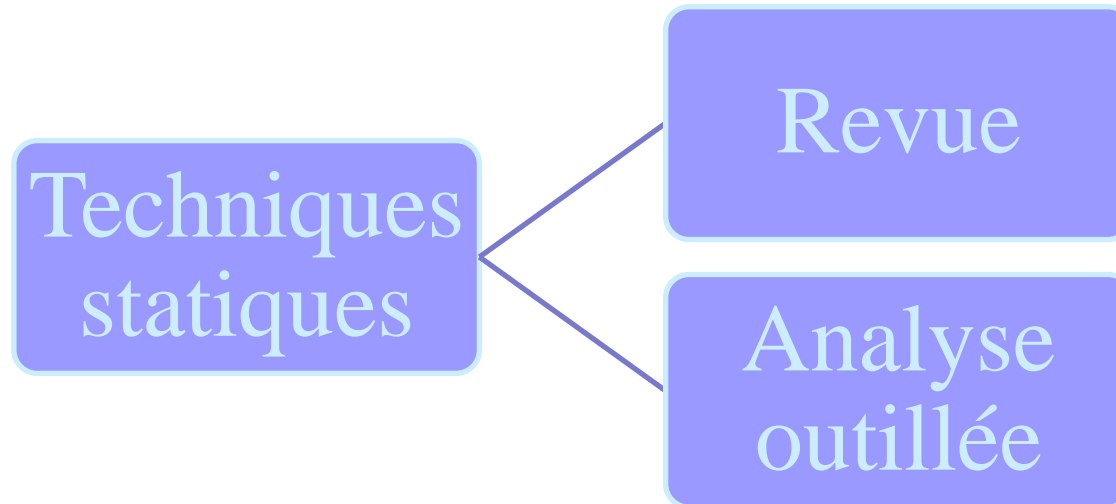
Mais, techniques complémentaires !

3.1 Techniques statiques et processus de test (K2)

- ◆ LO-3.1.1 Reconnaître les livrables qui peuvent être examinés par les différentes techniques statiques. (K1)
- ◆ LO-3.1.2 Décrire l'importance et la valeur de l'utilisation de techniques statiques dans l'évaluation de livrables logiciels. (K2)
- ◆ LO-3.1.3 Expliquer les différences entre les techniques statiques et dynamiques, en considérant les objectifs, les types de défauts à identifier et le rôle de ces techniques dans le cycle de vie du logiciel. (K2)

Techniques statiques et processus de test

- ◆ Les techniques statiques ne s'appuient pas sur l'exécution du code
- ◆ Deux types de techniques :



- ◆ S'appliquent sur l'ensemble des artefacts du développement logiciel : expression de besoins, spécifications, code, tests...

Principes des techniques statiques

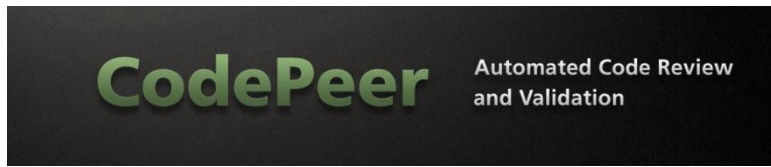
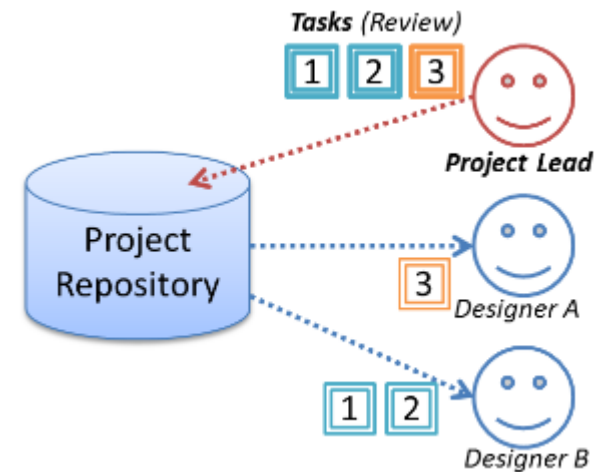
- ◆ Les revues sont une manière de tester des produits logiciels (y compris du code) et peuvent être exécutées bien avant l'exécution de tests dynamiques.
- ◆ Les défauts détectés pendant les revues effectuées tôt dans le cycle de vie sont souvent bien moins coûteux à ôter que les défauts détectés lors de l'exécution des tests (p.ex. défauts trouvés dans les exigences).
- ◆ Contrairement aux tests dynamiques, les techniques statiques trouvent les causes des défauts plutôt que les défaillances elles-même.
- ◆ Les défauts typiques plus faciles à trouver lors de revues que pendant les tests dynamiques sont : déviations par rapport aux standards, défauts d'exigences, défauts de conception, maintenabilité insuffisante et spécifications incorrectes d'interfaces.

Revue manuelles vs Outillage d'analyse

- ◆ Revue manuelle → activité collaborative de recherche de défauts dans un artefact du projet



- ◆ Analyse statique outillée → détection outillée de défaut dans un artefact formel (modèle, code, scripts, ...) + mesure de qualité



Objet de la revue

- ◆ Tous les artefacts de projet peuvent être revu :
 - les exigences métier et les documents d'expression de besoins (cas utilisation, processus métier, ...)
 - les spécifications de conception et les documents d'architecture
 - Les documents de test (stratégie, plan de test, cas de test, scripts)
 - Code de l'application
 - IHM
 - Documentation
 -

Apports des techniques statiques

- ◆ Détection anticipée des défauts et des omissions (par exemple dans les spécifications)
- ◆ Améliorations de la qualité du code (par exemple par le respect des standards, et des bonnes pratiques)
- ◆ Diminution du nombre de défauts (prévient la multiplication des anomalies)
- ◆ Meilleure communication entre membres de l'équipe

Quiz – section 3.1

Q1 – Lequel des éléments suivants peut-il être examiné avec une technique de revue ?

- a) Du code logiciel
- b) Des spécifications d'exigences
- c) Des conceptions de test
- d) Tous les éléments cités ci-dessus

Quiz – section 3.1

Q2 – Quelle est la phrase fausse ?

- a) Le test dynamique s'appuie sur l'exécution du logiciel
- b) Les tests statiques incluent les tests non fonctionnels
- c) Le test statique est sans exécution du logiciel
- d) Les revues sont un type de test statique

Quiz – section 3.1

Q2 – Quelle est la phrase fausse ?

- a) Le test dynamique s'appuie sur l'exécution du logiciel
- b) Les tests statiques incluent les tests non fonctionnels
- c) Le test statique est sans exécution du logiciel
- d) Les revues sont un type de test statique

Réponses au Quiz

◆ Section 3.1

- $Q1 \rightarrow d$
- $Q2 \rightarrow b$

3.2 Processus de revue (K2)

- ◆ LO-3.2.1 Rappeler les activités, rôles et responsabilités d'une revue formelle typique. (K1)
- ◆ LO-3.2.2 Expliquer les différences entre les différents types de revues: revue informelle, revue technique, relecture technique et inspection. (K2)
- ◆ LO-3.2.3 Expliquer les facteurs liés à l'exécution de revues couronnées de succès. (K2)

Processus de revue

Les différents types de revues varient de **informel** à **systématique** :

- **Informel**
 - Caractérisé par l'absence d'instructions écrites pour les réviseurs
- **Systématique**
 - Caractérisé par :
 - des procédures documentées pour mener la revue
 - la participation des parties prenantes
 - des résultats documentés de la revue

**Revue
Informelle**

**Relecture
Technique**

**Revue
Technique**

Inspection

- formel

+ formel

Termes (à retenir)



◆ Revue informelle

- une revue qui n'est pas basée sur une procédure formelle (documentée)

◆ Relecture technique

- Présentation pas à pas par l'auteur d'un document de façon à réunir des informations et à établir une compréhension commune de son contenu

Termes (à retenir)



◆ Revue technique

- Une activité de discussions de groupes de pairs qui se focalise sur l'obtention d'un consensus sur une approche technique à prendre.
Aussi appelée revue de pairs

◆ Inspection

- un type de revue qui se base sur un examen visuel de documents pour détecter des défauts (p.ex. violation des standards de développement et non respect de documentation de haut niveau).
Technique de revue la plus formelle et donc toujours basées sur des procédures documentées

Processus de revue

- ✓ La manière dont une revue est exécutée dépend des **objectifs** convenus pour la revue, par exemple:
 - trouver des défauts,
 - augmenter la compréhension,
 - former les testeurs et les nouveaux membres d'une équipe,
 - ou organiser la discussion et décider par consensus.

Phases d'une revue formelle

Une revue formelle comprend **six phases principales** :

- Distribuer les documents
- Expliquer les objectifs, le processus et les documents
- Examiner, Evaluer, discuter
- Noter les défauts
- Faire des préconisations
- Faire un CR de la réunion
- Vérifier que les défauts ont bien été traités
- Récolter les métriques
- Vérification des critères de sortie

Planification

Lancement

Préparation individuelle

Réunion de revue

Correction des défauts

Suivi

- Choisir le personnel, allouer les rôles
- Définir les critères d'entrées et de sortie (revue formelle)
- Sélectionner des parties du document à examiner
- Vérifier les critères d'entrée

- Préparer la réunion de revue en revoyant le(s) document(s)
- Identifier et rapporter les défauts potentiels, les questions et commentaires

- Corriger les défauts

Rôles et responsabilités

Une revue formelle typique inclura les 5 rôles principaux suivants :

- Réviseur,
- Manager,
- Modérateur,
- Scribe,
- Auteur.

Ces rôles sont décrits dans la suite.

Processus de revue – Rôles et responsabilité

Manager : décide l'exécution des revues, alloue le temps dans la planification du projet et détermine si les objectifs de revue ont été atteints.

Modérateur : la personne qui dirige la revue du document ou de l'ensemble des documents, incluant :

- la planification de la revue,
- l'exécution de la revue
- le suivi post-réunion.

Si besoin, le modérateur peut servir d'intermédiaire entre les différents points de vue et est souvent la personne sur qui repose le succès d'une revue.

Rôles et responsabilités

Auteur : l'auteur ou la personne à qui incombe la responsabilité principale du ou des document(s) à revoir.

Scribe : documente tous les aspects, problèmes et points ouverts identifiés pendant la réunion.

Réviseurs : les individus avec une culture technique ou métier spécifique (aussi appelés vérificateurs ou inspecteurs) qui, après la préparation nécessaire, identifient et décrivent les constatations (p.ex. les défauts) dans le produit en cours de revue.

Méthode de travail

Utilisation de check-lists

En examinant des documents selon différentes perspectives, et en utilisant des check-lists, les revues peuvent devenir plus efficaces et rentables.

Par exemple :

- une check-list basée sur la perspective :
 - d'un utilisateur
 - d'un mainteneur
 - d'un testeur
 - d'un opérateur
- une check-list reprenant des problèmes d'exigences typiques

Types de revue

- ◆ En allant du moins formel au plus formel, on considère 4 types de revue :
 - La revue informelle,
 - La relecture technique,
 - La revue technique,
 - L'inspection.

Types de revue

Revue informelle

- Objectif principal :
Manière bon marché d'obtenir des résultats
- Pas de processus formel
- Peut inclure la programmation par paires ou une revue de conception et de code par un responsable technique
- Les résultats peuvent être documentés
- Peut varier en utilité selon les réviseurs

En anglais, on parle de « buddy review » = revue entre copains

Types de revues

Relecture technique

- Objectifs principaux :
 - apprendre
 - gagner en compréhension
 - trouver des défauts
- Réunion dirigée par l'auteur
- Peut prendre la forme de scénarios, répétitions à blanc, participation de groupes de pairs
- Optionnellement une réunion de préparation de revue par les réviseurs
- Optionnellement préparation d'un rapport de revue incluant une liste de constatations
- Optionnellement un scribe (qui n'est pas l'auteur)
- Varie en pratique de quasiment informelle à très formelle

Types de revue

Revue technique (1/2) = Revue de pairs

- Objectifs principaux :
 - discuter, décider
 - évaluer des alternatives
 - trouver des défauts
 - résoudre des problèmes techniques
 - vérifier la conformité aux spécifications, plans, réglementations et standards
- Appelée aussi revue des pairs
- Peut varier en pratique de quasiment informelle à très formelle
- Peut optionnellement utiliser des check-lists

Types de revue

Revue technique (2/2)

- Documentée, processus de détection de défauts défini incluant des pairs et des experts techniques avec optionnellement la participation de l'encadrement
- Peut être effectuée comme une revue de pairs sans participation de l'encadrement
- Idéalement dirigée par un modérateur formé (pas l'auteur)
- Réunion de préparation par les réviseurs
- Préparation d'un rapport de revue incluant la liste de constatations, le verdict indiquant si le produit logiciel répond à ses exigences et, si appropriées, des recommandations relatives aux constatations ;

Types de revue

Inspection :

- Objectif principal : trouver des défauts
- Dirigée par un modérateur formé (pas l'auteur)
- Généralement menée comme un examen par les pairs
- Rôles définis
- Inclut des métriques
- Processus formel basé sur des règles et des check-lists
- Critères d'entrée et de sortie spécifiés pour l'acceptation du produit logiciel
- Réunion de préparation
- Rapport d'inspection incluant la liste de constatations
- un lecteur (facultatif)

Facteurs de succès des revues

Les facteurs de succès des revues incluent (1/2) :

- Chaque revue a des objectifs prédéfinis et clairs.
- Les personnes impliquées sont adéquates pour les objectifs de la revue.
- Les testeurs sont des réviseurs de valeur qui contribuent à la revue et ainsi prennent connaissance du produit afin de pouvoir préparer les tests plus tôt
- Les défauts trouvés sont bien acceptés, et exprimés objectivement.
- Les aspects personnels et psychologiques sont traités
p.ex. en faisant de cela une expérience positive pour l'auteur
- La revue est menée dans une atmosphère de confiance - Les résultats ne sont pas utilisés pour évaluer les participants

Facteurs de succès des revues

Les facteurs de succès des revues incluent (2/2) :

- Les techniques de revue adaptées aux objectifs, adaptées aux types et au niveau de livrable logiciel, et adaptées aux types et niveau des réviseurs, sont appliquées.
- Des check-lists ou des profils sont utilisés lorsque cela est approprié, afin d'augmenter l'efficacité de détection des défauts.
- Des formations sont données sur les techniques de revue, en particulier celles concernant les techniques plus formelles telles que les inspections.
- L'encadrement supporte un bon processus de revue (p.ex. en incorporant du temps pour les activités de revue dans les plannings des projets).
- L'accent est mis sur l'apprentissage et l'amélioration du processus.

Quiz – section 3.2

◆ Q1 – Quelles sont les correspondances entre les caractéristiques et types de revue ?

I- Dirigé par l'auteur

II- Non documenté

III- Pas de participation du management

IV- Dirigé par un modérateur ou manager formé

V- Fonctionne avec des critères d'entrée et de sortie

A- Inspection

B- Revue technique

C- Revue informelle

D- Relecture technique

Réponses:

a) A = IV, B = III, C = II et V, D = I

b) A = IV et V, B = III, C = II, D = I

c) A = I et V, B = III, C = II, D = IV

d) A = V, B = V, C = III, D = I et II

Réponses au Quiz

- ◆ **Section 3.2**
 - $Q1 \rightarrow b$

3.3 Analyse statique avec des outils (K2)

- ◆ LO-3.3.1 Rappeler les défauts et erreurs typiques identifiées par les analyses statiques et les comparer à ceux détectés par les revues et tests dynamiques. (K1)
- ◆ LO-3.3.2 Décrire, en utilisant des exemples, les avantages typiques des analyses statiques. (K2)
- ◆ LO-3.3.3 Lister les défauts typiques dans le code et la conception qui peuvent être identifiés par des outils d'analyse statique. (K1)

Termes importants (définition à retenir)



◆ Analyse statique

- Analyse des artefacts logiciels (par exemple, conception ou code), effectuée sans l'exécution de ces derniers

◆ Flux de contrôle

- Une représentation abstraite de toutes les séquences d'événements (chemins) dans l'exécution d'un composant ou système

◆ Flot de données

- Une représentation abstraite de la séquence et des modifications possibles de l'état des objets de données, où l'état d'un objet est soit création, utilisation ou destruction

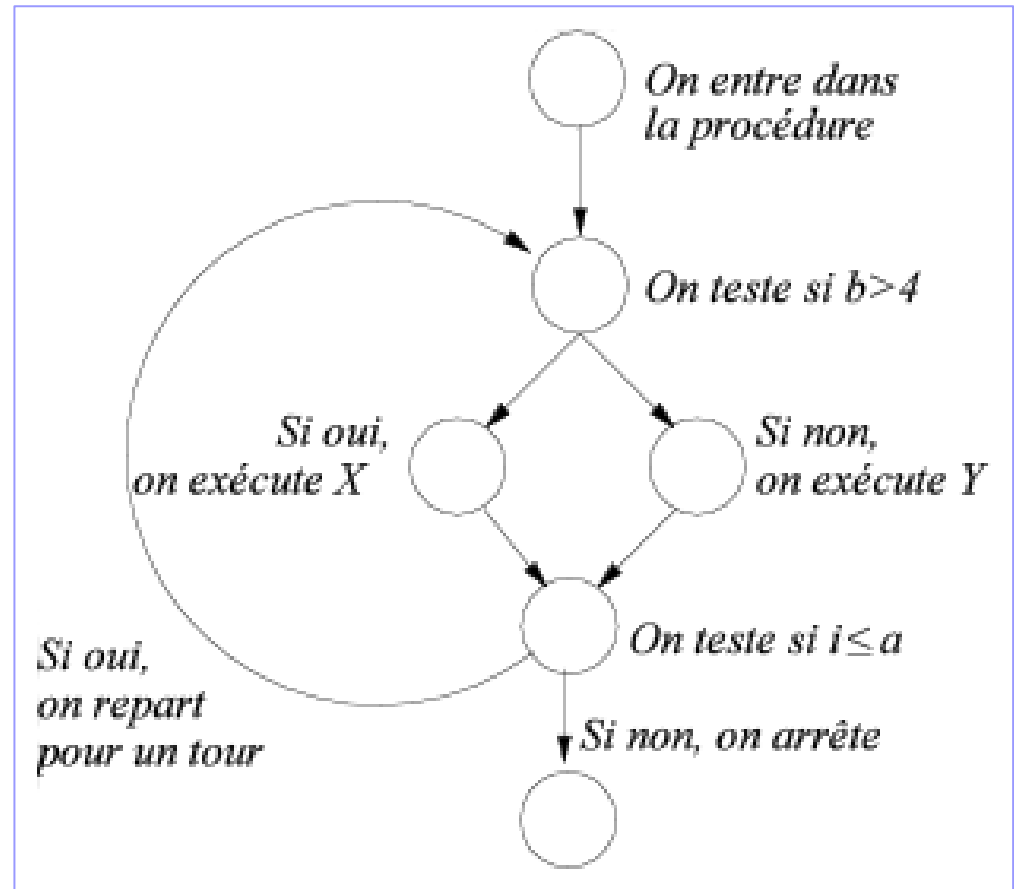
Types de défauts détectés par les outils d'analyse statique de code

- ◆ Référencement d'une variable avec une valeur indéfinie
- ◆ Interface inconsistante entre modules et composants
- ◆ Variables qui ne sont jamais utilisées ou déclarées de façon incorrecte
- ◆ Code non accessible (code mort)
- ◆ Logique absente et erronée (potentiellement des boucles infinies)
- ◆ Constructions trop compliquées
- ◆ Violation des standards de programmation
- ◆ Vulnérabilités de sécurité
- ◆ Violation de syntaxe dans le code et les modèles logiciels

Méthodes de l'analyse statique – 1/3

Analyse du graphe de flot de contrôle

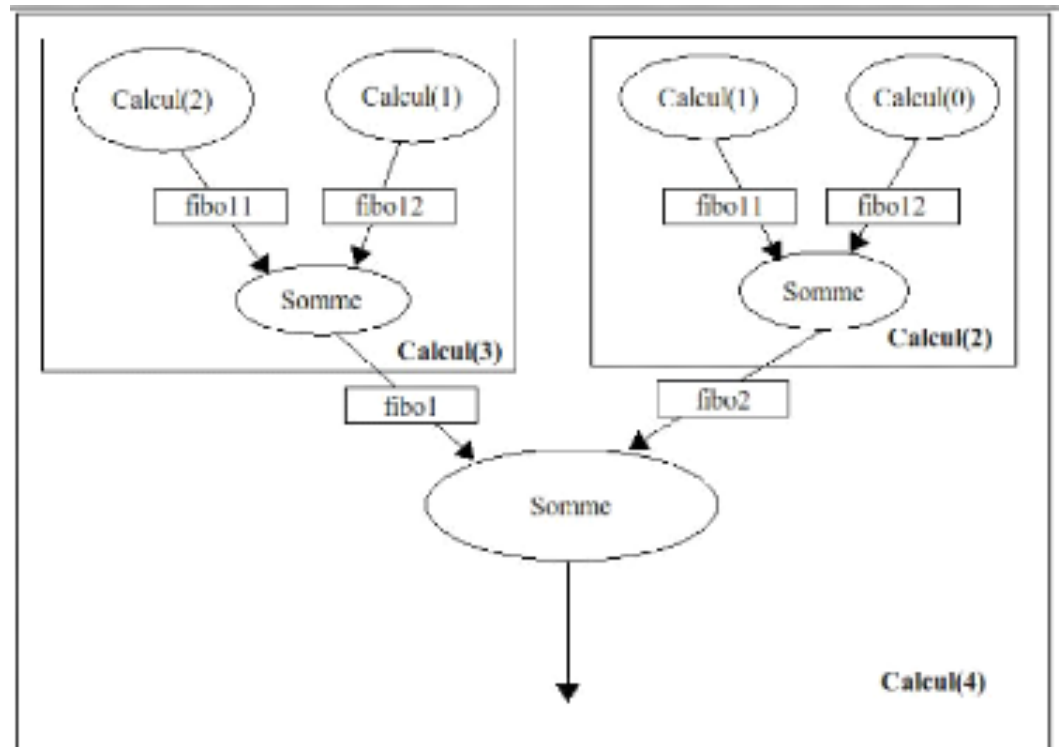
- ◆ Atteignabilité
- ◆ Boucles infinies
- ◆ Code mort
- ◆



Méthodes de l'analyse statique – 2/3

Analyse du graphe de flux de données

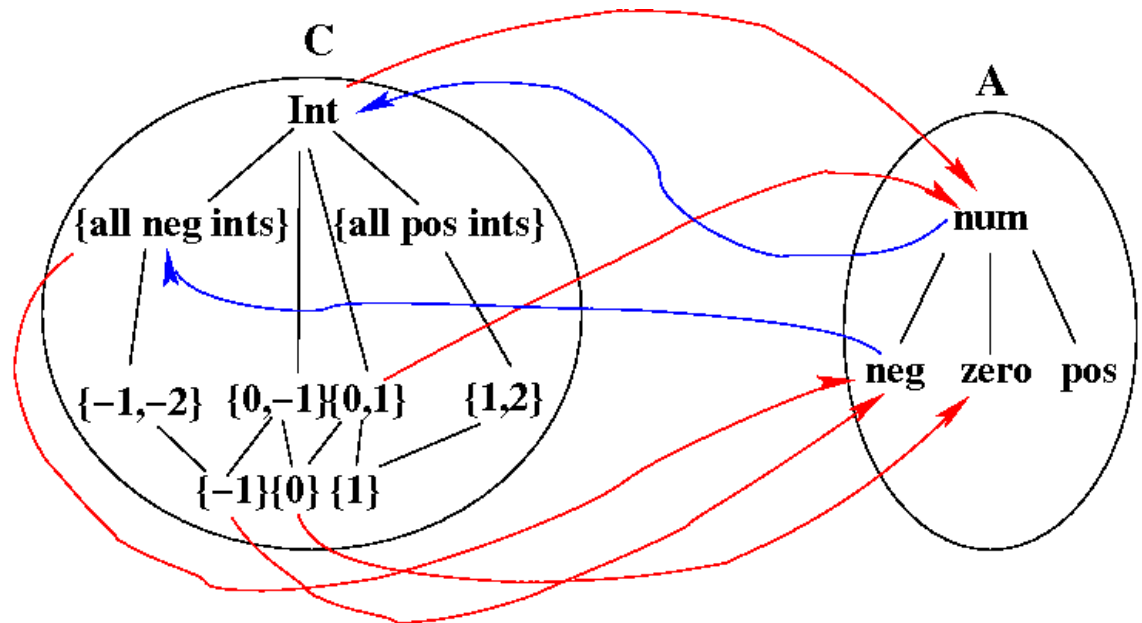
- ◆ Division par 0
- ◆ Débordement de pointeur
- ◆ ...



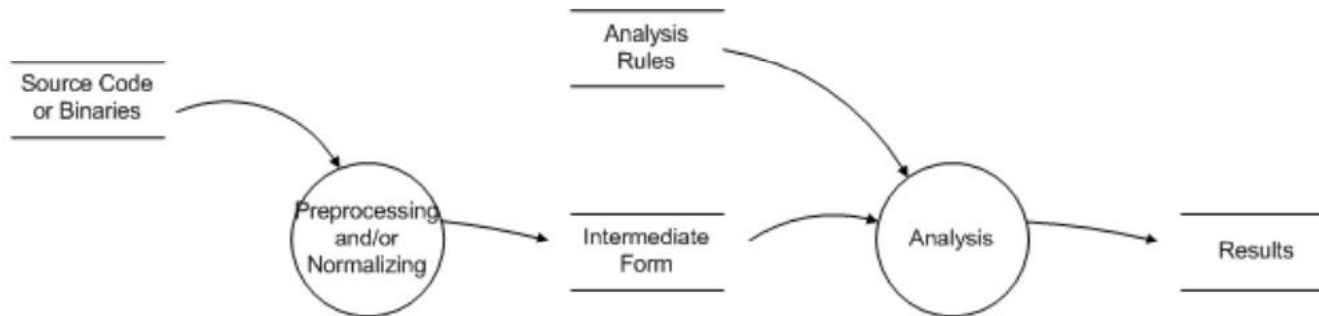
Méthodes de l'analyse statique – 3/3

Interprétation abstraite

- Exécution symbolique d'un programme (par propagation de contraintes)
- Abstraction du programme (par exemple ne manipuler que + / -)



Analyse statique de sécurité



Source :
<http://fr.slideshare.net/denimgroup/static-analysis-techniques-for-testing-application-security-houston-tech-fest>

- ◆ S'applique sur le source ou le binaire
- ◆ A base de règles d'analyse
- ◆ Cible certaines vulnérabilités OWASP Top 10 : SQL Injection, XSS, Buffer overflow
- ◆ Problème des faux positifs

Les bénéfices de l'analyse statique outillée

- ◆ La détection très tôt de défauts avant l'exécution des tests.
- ◆ Une information très tôt sur certains aspects suspects du code ou de la conception, par le calcul de métriques, par exemple une mesure de complexité élevée.
- ◆ L'identification de défauts difficilement détectables par des tests dynamiques.
- ◆ La détection de dépendances et d'inconsistances dans les modèles logiciels tels que des liens dans les modèles logiciels
- ◆ L'amélioration de la maintenabilité du code et de la conception.
- ◆ La prévention des défauts, si les leçons sont prises en compte lors du développement.

Outils d'analyse statiques

- ◆ On trouve des fonctions d'analyse statique de code dans les outils de développements standards :
 - Compilateurs
 - Environnement de développements
- ◆ Il existe de nombreux outils, en particulier dédiés à un langage particulier et/ou à un type de détection (par exemple sur les vulnérabilités de sécurité)
- ◆ Exemples :
 - Sonargraph – cf <http://www.hello2morrow.com/products/sonargraph>
 - FindBugs – cf <http://findbugs.sourceforge.net/>

Voir : http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis

3.3 Analyse statique outillée (K2)

- ◆ LO-3.3.1 Rappeler les défauts et erreurs typiques identifiées par les analyses statiques et les comparer à ceux détectés par les revues et tests dynamiques. (K1)
- ◆ LO-3.3.2 Décrire, en utilisant des exemples, les avantages typiques des analyses statiques. (K2)
- ◆ LO-3.3.3 Lister les défauts typiques dans le code et la conception qui peuvent être identifiés par des outils d'analyse statique. (K1)

Quiz – section 3.3

Q1 – Quelle est la phrase juste ?

- a) L'analyse statique de code permet de détecter automatiquement les erreurs de spécifications
- b) L'analyse statique de code permet l'identification de défauts difficilement détectables par des tests dynamiques.
- c) L'analyse statique de code n'est utile que pour le test de vulnérabilités de sécurité
- d) L'analyse statique outillée ne peut porter que sur le code

Réponses au Quiz

- ◆ **Section 3.3**
 - $Q1 \rightarrow b$