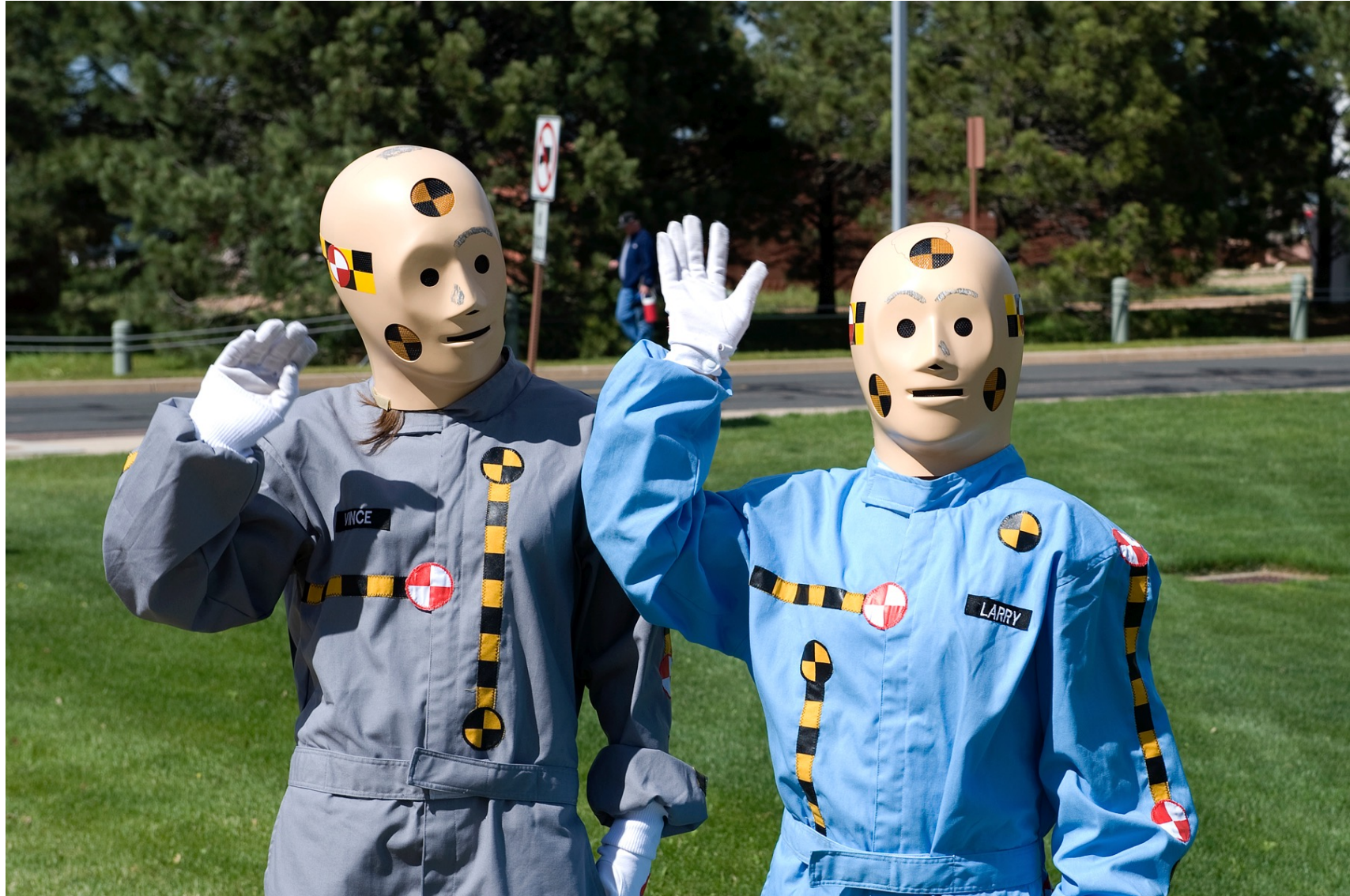


Comment écrire de bons tests ?



Right ?

- validation des résultats en fonction de ce que définit la spécification
 - on doit pouvoir répondre à la question « comment sait-on que le programme s'est exécuté correctement ? »
 - si pas de réponse => spécifications certainement vagues, incomplètes
- Utilise plutôt une valeur commune :
 - Racine carrée ?
 - $\sqrt{4} \Rightarrow 2$

B : Boundary conditions

- Les premiers à faire après le test Right
- Ceux qui ont le plus de valeur !
- Des bornes ? Des conditions limites ? Partout !
 - L'âge d'une personne ?
 - L'email d'un étudiant ?
 - Le nombre de place dans un train ?
 - L'URL d'accès à ce fichier ?
 - $v_0 \Rightarrow ???$
- Boundaries must be **CORRECT**

Correct: Conformance

- Test avec données en dehors du format attendu
- Un email ?
 - anachroniques ex. : !*W@\"
 - non correctement formatées ex. : fred@foobar.
 - pas souhaitées : pouetpouet@yopmail.fr

cOrrect: Order

- Test avec données sans l'ordre attendu
 - Les éléments sont censés être dans un certain ordre, que se passe-t-il s'ils le sont pas ?

coRrect: Range

- Un intervalle quelque part ?
- Un indice dans un tableau
- L'indice est inférieur à 0...
- L'indice est supérieur à la taille max...

corRect: Reference

- Quelles sont les hypothèses de la classe que l'on teste ?
 - Un objet est toujours présent ou dans un certain état lorsqu'il est passé en paramètre au constructeur ou à une méthode... Et sinon ?

corrEct: Existence

- Que se passe-t-il si une valeur n'existe pas ?
- Que tourne `stack.pop()` ?
 - Null
 - Une exception ? Laquelle ?
 - Un objet bizarre qui représente la pile vide ???

correCt: Cardinality

- Y-a-t-il un comptage de valeurs ?
- Un nombre d'éléments à vérifier avec un calcul ?
- Combien de poteaux pour faire une barrière de 20m avec 1 poteau tous les 2 mètres ?
- **Règle du 0-1-N**

correct: Time

- **Ordering Time**

- Une méthode après forcément après une autre ?

- **Absolute Time**

- Timezone

- DST ? Passage à l'heure d'été, d'hiver

- Un rattrapage de secondes ?

Inverse

- Identifie **les relations inverses** qui permettent de vérifier le comportement
- Test de racine carrée :
 - $\sqrt{a^2} = a$
 - Avec une valeur quelconque de a (mais pas 1...)

Cross check

- Identifie **les algorithmes équivalents** qui permettent de vérifier le comportement
- Test de racine carrée :
 - $\sqrt{474} = \text{Math.sqrt}(474)$

Ma version



La version équivalente



Error condition

- Identifier ce qui se passe quand
 - Le disque, la mémoire sont pleins
 - Il y a perte de connexion réseau
 - Un argument n'est pas dans le domaine attendu
- Equivalent à la notion de robustesse
- Exemple :
 - $\sqrt{-1} \Rightarrow$ Illegal Argument

Performance

- Vérifier des performances attendues
 - $\text{Time}(v474) < 200 \text{ ms}$
- Identifier des régressions dans les performances
- Attention, cette partie est un domaine de test non-fonctionnel à part entière (charge, performance, etc.).

Synthèse

- Right : est-ce que les résultats sont corrects ?
- B (Boundary) : est-ce que les conditions aux limites sont correctes ?
- I (Inverse) : est-ce que l'on peut vérifier la relation inverse ?
- C (Cross-check) : est-ce que l'on peut vérifier le résultat autrement ?
- E (Error condition) : est-ce que l'on peut forcer l'occurrence d'erreurs ?
- P (Performance) : est-ce que les performances sont prévisibles ?