



Les tests logiciels

Par Martin Dubois, ing.

2015-07-08

KMS - Martin Dubois, ing.

1

Plan de la présentation



- Présentations du formateur et des participants
- Est-il possible de **tout tester**?
- Que faut-il tester?
- Est-il nécessaire d'**automatiser** les tests?
- Comment **automatiser** les tests?
- **Quand** est-il préférable d'exécuter les tests?
- Période de questions

2015-07-08

KMS - Martin Dubois, ing.

2

Partie 1

EST-IL POSSIBLE DE TOUT TESTER?

2015-07-08

KMS - Martin Dubois, ing.

3

Que signifie « Tout tester »?

- Est-ce qu'une **couverture** du code de 100 % signifie que tout est testé?

```
printf("Reussi: %u / %u - %f%%\n"  
      okCount, opCount,  
      static_cast<double>(okCount) * 100.0 /  
      static_cast<double>(opCount));
```

2015-07-08

KMS - Martin Dubois, ing.

4

La réponse théorique

- Oui, dans un monde idéal il est théoriquement possible de tout tester!
- Mais vivons-nous dans un monde idéal?
- Le temps disponible avant de livrer est-il « Idéal »?
- Les ressources sont-elles illimitées?

La réponse courte

NON!

Partie 2

QUE FAUT-IL TESTER?

2015-07-08

KMS - Martin Dubois, ing.

7

Les sources ou les exécutables

Code.c

```
int Add(int a, int b)
{
#ifdef _TEST_
    return 4;
#else
    return (a-b);
#endif
}
```

Test.c

```
#define _TEST_
#include "Code.c"
int main()
{
    if (4 == Add(2,2))
        printf("Pass!");
    else
        printf("Fail!");
}
```

2015-07-08

KMS - Martin Dubois, ing.

8

Les sources ou les exécutables?

- J'avoue! C'est un peu simpliste!
- Chose certaine, les exécutables dépendent des sources, mais aussi
 - Du compilateur
 - De la configuration du compilateur et du projet
 - Du mode de compilation (debug/release)
 - ...

Bref, il faut tester les exécutables!