Connaitre les outils de Design Control

· **Cadre général**

Le but être efficace lorsque l’on veut faire un design contrôle compliqué pour coordonner les ingénieurs

Convergence des outils. Le but

-Tenir les besoins clients (identifier ce que l’utilisateur veut, ce qui est un requis indispensable)

-Avoir une façon de tester si oui ou non j’ai atteint mes objectifs à chaque étape, un débuggage à chaque étape et pas seulement en fin de projetà vérification

-Garantir la sécurité de l’utilisateur : je dois être capable de prouver que mon design est fiable pour l’utilisateur. Pour le garantir il faut avoir des méthodes d’évaluation de probabilité et remonter ainsi aux marges autorisées sur les pièces.

· **Objectif de la formation**

URS (User Requirements) entre marketing et ingénieur : noir sur blanc ce que veut l’utilisateur.

SRS (System Requirements) approche boîte noire, j’écris exactement ce que mon système doit être capable de faire. Je me pose les questions sur les contraintes environnementales, réglementaires… Que veulent les utilisateurs et tous les autres acteurs.

SSRS sous-SRS je découpe en sous partie et je refais des boîte noires sur ce que le robot doit avoir. Le système doit lorsqu’il reçoit ceci cela donner en sortie cela.

TP test plan la façon dont je vais tester mon système pour vérifier qu’il rempli les exigences du SRS, on a même intérêt à prendre des marges sur les besoins du srs pour être sûr. Plus une erreur est détecté tôt plus c’est facile à corriger (le mieux étant de la détecter au moment de la tdr !) Non ambigüe : le robot passe ou échoue le test. Rq : tester même les trucs évidents pour ne pas les oublier

TR test result : j’ai testé, voilà le résultat.

Validation : faire vraiment des tests avec l’utilisateur, expérimentation en condition réel

Traçabilité : en face de chaque besoin j’ai mis une spécification, qui a été validée par un test.

FMEA et FTA

· **Exercice**

Table de contraste en médecine

On peut mettre du marrant genre Wahou effect dans URS plus dans SRS (ex : the robot shall look nice and modern)

· **URS**

Le gars du marketing ne doit pas se laisser trop influencer sur la méthode de solution. Le client va avoir des idées en têtes mais ce ne sont pas forcément les mieux.

· **Requirement : un art (dans URS SRS et SSRS)**

On définit le minimum nécessaire, atomique, chaque requirement est une seule chose non ambigüe ; On prend en compte les compromis entre les différents besoins (cohérence) un requirement est vérifiable.

· SRS

Les exigences fonctionnelles, non fonctionnelles (sécu, accessibilité…), les contraintes (livraison à l’étranger), exigences d’interface

Ne pas hésiter à mettre des schéma et des définitions pour ne pas avoir des requirements super longs.

Se fait en deux heures e puis on la soumet et une heure de màj.

· SSRS

On n’est plus au niveau du système mais du composant. Se rédige comme une SSRS juste changement de cadre

· TP et TR

Un cousin de 12 ans doit pouvoir faire le test, tableau avec :

(Type) , test action (ce qu’on fait), id (traçabilité), résultat attendu, résultat observé, conclusion. Ne pas introduire de mot ambigu sans le définir (par exemple : *précision*)

Le TP est figé, on ne le touche pas, on fait plusieurs TR avec des dates et on les archives jusqu’à un TR avec tout bon ! Se fait en deux heures et puis on la soumet et une heure de màj.

TR on peut garder des vidéos pour faire des RETEX

Ne pas faire des TR en permanence, avoir le bon dosage, d’en faire régulièrement permet de ne pas tourner en rond.

· Validation

On fait les tests avec l’utilisateur qui dit s’il valide ou non, si les TP-TR on bien étaient faits ça doit bien se passer normalement…

· Risks

Lister les risques avec leur sévérité et la probabilité de dégât ( le robot tombe, avec quelle proba il tue quelqu’un)

· FMEA (Failure Modes and Effects Analysis)

From failure to risk : analyse complète des éléments du systèmes, tous les mauvais comportement des pièces

ALARP : as low as reasonably possible : je ne peux pas faire plus sûr avec ce qu’il existe à l’heure actuelle (pas des questions de sous ou de flemme de faire mieux)

· FTA (Fault Tree Analysis)

Je pars du risque et je regarde ce qui peut le provoquer. Plus rapide mais moins exhaustif que la FMEA. Dans tous les cas commencer par lister les risques. Prends en compte les mauvaises utilisations des utilisateurs et pas que sur les composants internes du système.

· Traçabilité

Cycle en V

· **TDR technical design review**

Format plus libre le but présenter une solution à des experts pour en discuter. Elle doit résoudre un problème. Répond à la question j’ai fais ce choix là mais pourquoi ?

Input (à ne pas rater)

Design

Pour et contre

Comparaison de solutions

Conclusion

(celle sur l’exemple environ 16 slides avec des slides de titre)

Une demi-journée en équipe

Sur la TDR architecture ne pas oublier de mettre en valeur le côté motorisation indép de l’encodage.

C’est la phase où il ne faut pas hésiter à faire des croquis, des modélisations, même vague pour savoir la viabilité de ce que l’on veut faire en encombrement etc.