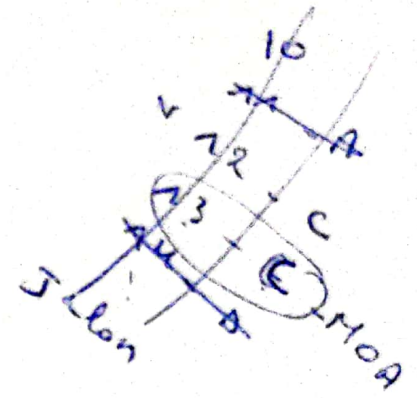


→ Définition Système :

c'est un ensemble d'éléments interagissant entre eux selon certains principes et/ou règles.



→ 2 types de systèmes :

- * Système à faire (système étudié) : à dominante technologique.

Ensemble organisé de matériel, logiciels, compétences humaines - pour répondre à un besoin donnée.

- * Système pour faire (projet) : à dominante organisationnelle.

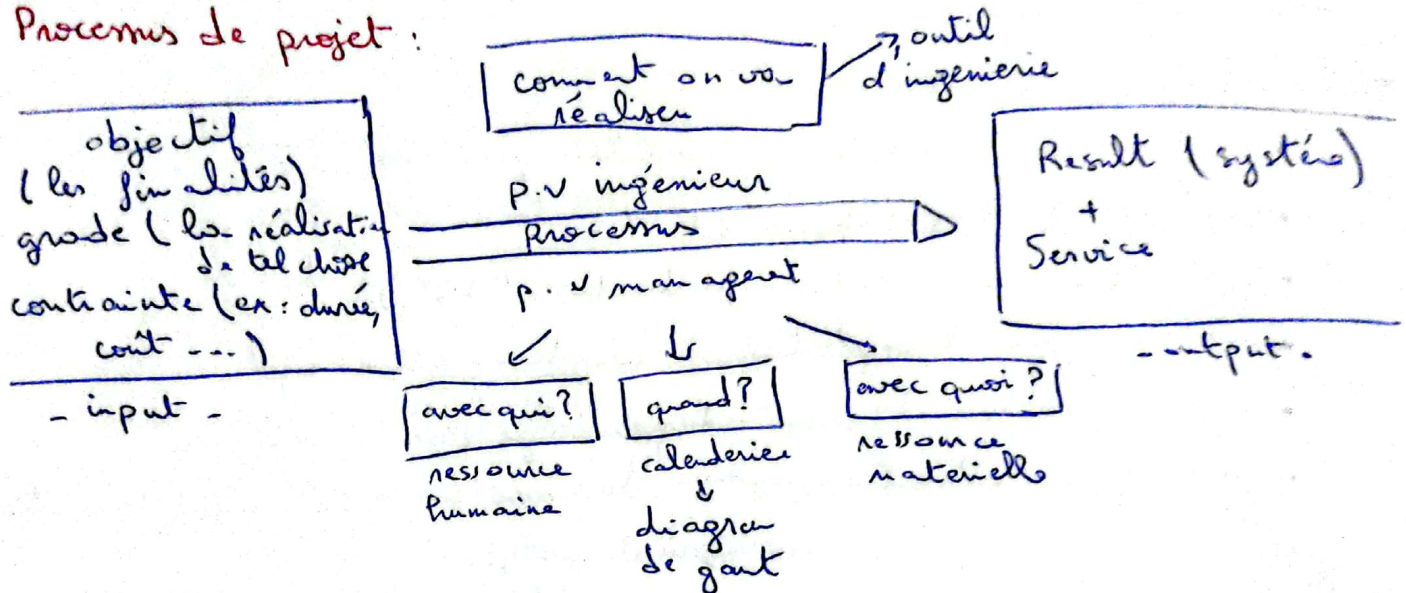
Ensembles coordonnées d'équipes, de méthode, de processus pour répondre à un objectif (ici le besoin de développer un système, le déployer, le faire évoluer)

- * un programme est un ensemble coordonnées de projets.

Faible IS:

- Retard
- dépassement de budget.
- confusion des exigences.
- non respect des contraintes techniques.
- conflit entre les équipes.
- les nerfs à vif.

Processus de projet:



Kickoff: commencement du projet

go life: fin du projet.

MOA: maître d'ouvrage (boîte noire) Client: partie prenante intéressé par l'utilisation (donne spécifications)

MOE: maître d'œuvre (boîte blanche) les développeurs: partie prenante impliquée dans la réalisation (test du système aussi).

ATOA: assistance de maîtrise d'ouvrage, aider l'utilisateur à rédiger son cahier de charge (besoin) par exemple.

peer review: pour éviter la mal compréhension entre les étapes du projet.

* vision boîte noire: analyser le système en interaction avec son environnement (specification fonctionnel par exemple).

* vision boîte blanche: analyse l'interaction de diff composants du système (conception, dev, production, test ...)

audite: par exemple audite de sécurité, audite de finance (vérifier tous les documents d'une société en cas de trafic ou qdq chose de pas normal).
opifisquer le code: changer le code de façon à ce que seul le développeur peut le comprendre (dans le cas où le contrat ne n'inclue pas le code source).

→ **audite de sécurité**: inspecter un système afin de trouver des failles de sécurité.

⇒ **Système Contributeurs**:

- système de conception: modelio par exemple.
- système de développement:
- système de réalisation: production (IDE)
- système pour l'exploitation: windows par exemple.
- système de maintenance: dépannage car de véhicule.
- système de retrait: machine de casse car de véhicule (désinstaller car informatique).

⇒ **Partie prenante (stakeholders)**:

- partie prenante intéressées par l'utilisation (MOA)
- " " impliquées dans la réalisation (MOE)
- " " potentiellement concernées (syndicats, associations)

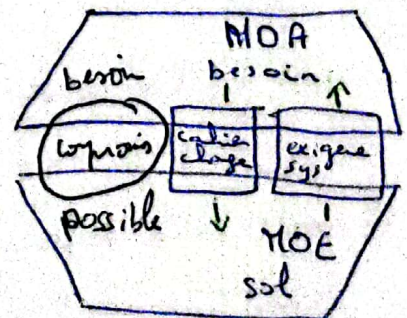
↳ **organisme de certification**:

iso 27001: certification des sécurité des systèmes.

iso 9001: gage de qualité, le processus de fabrication du produit est de qualité (pas forcément que le produit soit de qualité) faut rédiger les processus (fabrication, retour de produit, vente ...).

⇒ **Compromis entre besoins et contraintes**:

MOE: expriment les contraintes technique...



* Etude amont :

→ finalité

→ opportunité (esq la finalité est pertinente la réalisation est importante)

→ conceptualisation : qu'esq on va réaliser (les missions).

→ faisabilité : esq ce système est faisable.

* Analyse de l'existant :

→ Système existant à remplacer, modifier, ou à automatiser.

→ Environnement d'exploitation.

→ opportunité d'utilisation ou de réutilisation des éléments existants.

* Recueil des besoins :

ROI : retour on investissement

→ Repartitionner les parties prenantes.

→ identifier les personnes qui représentent les catégories.

→ intégration du besoin.

→ chercher un conflit entre besoin contradictoire (concurrents)

* Analyse des risques :

→ analyser les risques liés au système ainsi que les risques liés au projet (expérience par exemple ...) (compensé de conception exple)

→ identifier ces risques, puis pour chaque événement indésirable on identifie les actions possibles.

* Modèle de données exigées :

→ Emise par une source.

→ présente un ou plusieurs risques.

→ liste des modifications.

→ classé sur une catégorie.

→ vérifié par une méthode de vérification.

→ résultat d'un compromis.

→ justifié par une justification.

analyser fct sys :

fct rendu par sys : conditions de déclenchement, enchaînement des éléments de son mécanisme, Nature des flux échangés, Scénario d'échange, transformation à réaliser, no de service attendue

(type de rep, capacités, qualité de sortie),

incertitude et risque.

Conception:

* décomposition fonctionnelle puis physique : on commence par le fonctionnel car on a déjà les fonctionnalités définies puis chaque fct est alloué à des organes (physique) qui les réalise.

* ne composer (définir les interactions entre composants) architecture

=> une case fait partie de spécification fonctionnelle.

L> diagramme de séquence système (boîte noire)

L> " " " " objet (boîte blanche)

L> " " " " de classe (boîte blanche)

SOA: Service orienté architecture + des outils qui respecte SOA pour faire la communication avec plusieurs composants hétérogènes.

=> Architecture du système:

{ -> décomposition c'est un arbre (père et ses fils)

{ -> Architecture c'est un graphe avec relation entre nœuds.
(chaque flèche à un nom)

{ -> physique: les nœuds c'est des nom, fonctionnel les nœuds c'est des verbe (fct).

* vérifier: tester par rapport au spec fonctionnel.

* valider: " " " " au cahier de charge.

* qualification: Montrer si le sys est apte à répondre aux besoins, opérationnels (test à la réalité: répondre au vrai besoin).

* validation dans l'informatique: user acceptance test UAT

=> Méthode de vérification:

- inspection: relecture du code [ex: peer review]

- analyse théorique: simulation, modélisation

- démonstration: observation du sys en fonctionnement [inspection de tous le sys] sans instrument

- test et essais: exploitation des données acquies par instrumentation spécifique
JUPET par ex.

⇒ Change request: user veut modifier / ajouter un truc.

* Toute évolution détectée doit faire l'objet d'une étude d'impact

* entraîne un rebouclage sur la conception.

* tester tous le logiciel après la modif.

↳ vérification de non régression.

* il est conseillé de regrouper les évolutions impactant un constituant ou un nu de sys.

⇒ KPI: Key Performant Identification. (métrique).

* en relation avec le projet: exemple: équipe de pilotage a besoin des données (par ex: le taux d'avancement).

* en relation avec le sys à réaliser

* identification et suivi.

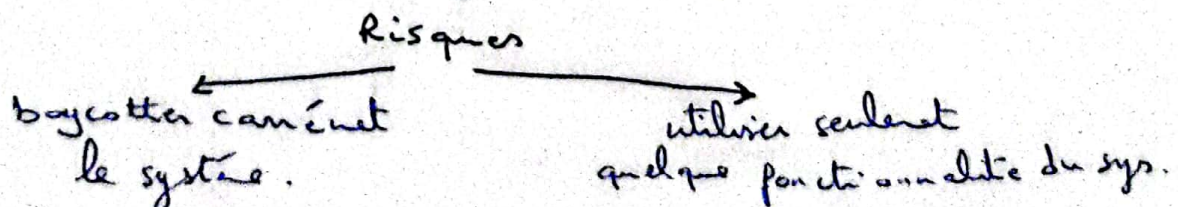
⇒ Méthode d'identification des métrique:

→ identification des risque ⇒ faire ressortir les raisons
(comme étant une donnée pour pouvoir gérer le projet.
→ pour éviter les risques.

KPI et identification des risque sont fortement relié.

⇒ Change Management:

↳ Résistance au nu sys: résistance au changement.



⇒ Solution:

→ Communication

→ formation des users: en cas besoin appel l'équipe de support technique.

→ Construire l'équipe de guidage.

* champions users: identifier des users compétant dans le sys déployé
| peuvent aider dans le change management.

↳ installation pilote : essai sur une tranche.

* le suivi : eq change management est bien effectué à l'aide des KPI (on ressort des statistique sur le taux d'utilisation du sys par exemple).

Jalons : une étape importante à marquer (finalisation d'une phase, commencer une phase ...).