

# Maîtrise des Grands Projets Informatiques

H1-H2  
Design

1. ~~~~~
2. ~~~~~
3. ~~~~~

H1-Headline

Video  
module

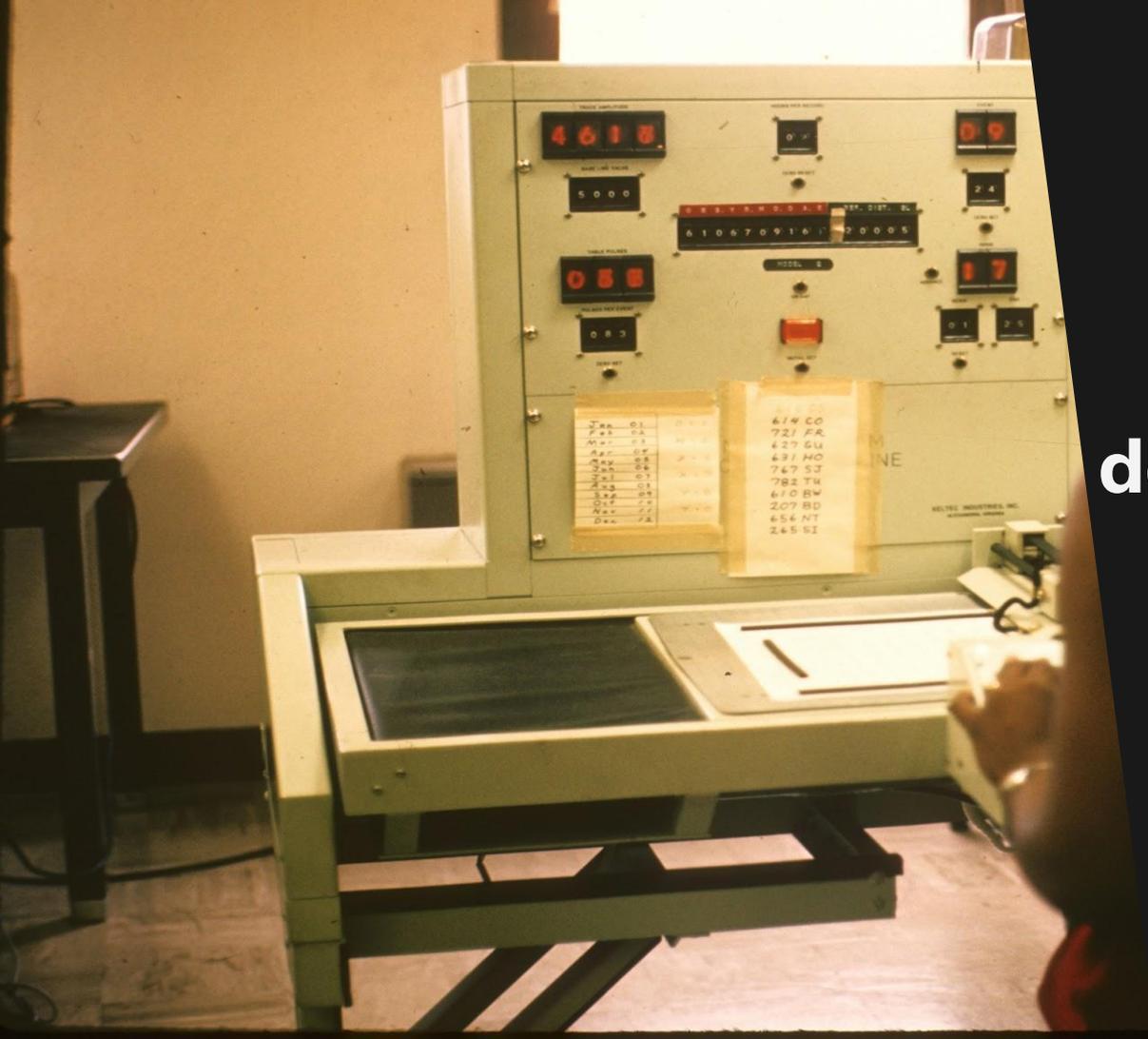
Description

Menu



# Plan du cours

- Introduction aux Concepts de Base de l'Organisation des Entreprises
- Cycle de vie et cycle de développement logiciel
- **Activités de développement logiciel**
- Avant-projet
- Collecte des besoins : cas d'utilisation, user's story
- Analyse Conception
- Codage
- Intégration
- Déploiement et production : plateforme de déploiement continu, surveillance des applications, conteneurisation en production (docker)
- Activités de gestion de projets



# Activités de développement logiciel

Jan	01	0001
Feb	02	0002
Mar	03	0003
Apr	04	0004
May	05	0005
Jun	06	0006
Jul	07	0007
Aug	08	0008
Sep	09	0009
Oct	10	0010
Nov	11	0011
Dec	12	0012

614	CO
727	FA
427	GU
437	HO
767	SJ
782	TU
610	BW
207	BD
654	NT
245	SI

# Activité de développement logiciel

Le développement logiciel est un processus complexe qui implique plusieurs étapes et activités pour créer des logiciels fiables, efficaces et répondant aux besoins des utilisateurs.

# Étapes du développement logiciel

- Planification
- Analyse des Besoins
- Conception
- Codage
- Tests
- Déploiement
- Maintenance

# Planification

La planification est la première étape du développement logiciel. Elle consiste à définir les objectifs du projet, à identifier les ressources nécessaires, à établir un calendrier et à déterminer les coûts. Une bonne planification permet de minimiser les risques et d'assurer que le projet est réalisable dans les délais et le budget impartis.

## Analyse des besoins

L'analyse des besoins consiste à comprendre et à documenter les exigences des utilisateurs. Cela inclut la collecte d'informations sur les fonctionnalités souhaitées, les contraintes techniques et les attentes des utilisateurs. Les analystes travaillent en étroite collaboration avec les parties prenantes pour s'assurer que toutes les exigences sont clairement définies et comprises.

## Conception

La conception est l'étape où les développeurs créent des plans détaillés pour le logiciel. Cela inclut la conception de l'architecture du système, la définition des interfaces utilisateur, la modélisation des données et la planification des algorithmes. Une bonne conception permet de garantir que le logiciel est modulaire, évolutif et facile à maintenir.

# Conception

La conception est l'étape où les développeurs créent des plans détaillés pour le logiciel. Cela inclut la conception de l'architecture du système, la définition des interfaces utilisateur, la modélisation des données et la planification des algorithmes. Une bonne conception permet de garantir que le logiciel est modulaire, évolutif et facile à maintenir.

## Codage

Le codage est l'étape où les développeurs écrivent le code source du logiciel. Ils utilisent des langages de programmation pour implémenter les fonctionnalités définies dans la phase de conception. Le codage doit être effectué de manière rigoureuse et conforme aux bonnes pratiques de développement pour garantir la qualité et la maintenabilité du code.

# Tests

Les tests sont une étape essentielle pour vérifier que le logiciel fonctionne comme prévu. Les tests peuvent être de différents types, tels que les tests unitaires, les tests d'intégration, les tests système et les tests d'acceptation. Les tests permettent de détecter et de corriger les bugs avant que le logiciel ne soit déployé.

# Déploiement

Le déploiement consiste à mettre le logiciel en production. Cela inclut l'installation du logiciel sur les serveurs, la configuration des environnements de production et la mise à disposition du logiciel aux utilisateurs finaux. Le déploiement doit être soigneusement planifié et exécuté pour minimiser les interruptions de service.

## Maintenance

La maintenance est l'étape où le logiciel est surveillé et mis à jour après son déploiement. Cela inclut la correction des bugs, l'ajout de nouvelles fonctionnalités et l'amélioration des performances. La maintenance est essentielle pour garantir que le logiciel reste fonctionnel et répond aux besoins changeants des utilisateurs.

# Plan du cours

- Introduction aux Concepts de Base de l'Organisation des Entreprises
- Cycle de vie et cycle de développement logiciel
- Activités de développement logiciel
- **Avant-projet**
- Collecte des besoins : cas d'utilisation, user's story
- Analyse Conception
- Codage
- Intégration
- Déploiement et production : plateforme de déploiement continu, surveillance des applications, conteneurisation en production (docker)
- Activités de gestion de projets



**Avant-Projet**

## Avant-projet

L'avant projet est une étape préliminaire au démarrage d'un projet. Durant cette étape, plusieurs éléments seront étudiés afin d'étudier la faisabilité, identifier les risques et définir clairement les objectifs du projet.

## Avant-Projet: Pourquoi ?

**70%** des projets ont tendance à échouer.

Selon Pulse of the profession 2018 du Project Management Institute (PMI) effectué auprès de 5402 entreprises les causes principales sont:

- un changement dans les priorités de l'entreprise (39%) ;
- un changement dans les objectifs du projet (37%) ;
- une collecte erronée des exigences (35%).

## Évaluation de la Charge

L'évaluation de la charge consiste à estimer les ressources nécessaires pour mener à bien le projet. Cela inclut l'estimation du temps, des coûts et des compétences requises.

## Avant-projet

Suivant le mode de gestion de projet, l'avant-projet peut prendre plusieurs formes.

Dans le cas d'utilisation des méthodes agiles, cela peut consister à s'assurer que les équipes sont formées à certaines technologies, découper le projet en plusieurs phases, ou vérifier la disponibilité d'experts devant intervenir sur le projet.

# Gestion des risques

La gestion des risques consiste à identifier, analyser et atténuer les risques potentiels qui pourraient affecter le projet.

Les risques peuvent être de plusieurs natures:

- Juridiques et réglementaires
- Financiers
- Risques opérationnels
- Etc

# Méthodes de planifications

Des méthodes afin de faciliter chacune des étapes d'avant-projet existent, on peut citer parmi les plus courantes:

- Le brainstorming
- L'analyse SWOT
- Les matrices de risques

# Brainstorming

Le déroulé d'un brainstorming se déroule généralement comme suit:

- Définir clairement l'objectif de la session de brainstorming.
- Encourager les participants à proposer des idées sur les risques potentiels. Toutes les idées sont notées sans jugement.
- Après la génération d'idées, clarifier et regrouper les idées similaires.

## Analyse SWOT

L'analyse SWOT (Strengths, Weaknesses, Opportunities, Threats) est une méthode d'analyse stratégique qui permet d'évaluer les forces, faiblesses, opportunités et menaces d'un projet.

## SWOT: Forces

Les aspects internes positifs du projet, comme les compétences de l'équipe, les ressources disponibles, ou les avantages concurrentiels.

## **SWOT: Faiblesse**

Les aspects internes négatifs du projet, comme les lacunes en compétences, les contraintes budgétaires, ou les dépendances technologiques.

## SWOT: Opportunités

Les aspects externes positifs qui peuvent être exploités, comme les tendances du marché, les nouvelles technologies, ou les partenariats potentiels.

## SWOT: Menaces

Les aspects externes négatifs qui peuvent affecter le projet, comme la concurrence, les changements réglementaires, ou les risques économiques.

# SWOT

	Positif (pour atteindre l'objectif)	Négatif (pour atteindre l'objectif)
Origine interne (organisationnelle)	<b>Forces</b> <b>S</b> Strengths	<b>Faiblesses</b> <b>W</b> Weaknesses
Origine externe (origine = environnement)	<b>Opportunités</b> <b>O</b> Opportunities	<b>Menaces</b> <b>T</b> Threats

## Matrice des risques

La matrice des risques est un outil de gestion des risques qui permet de classer les risques en fonction de leur impact et de leur probabilité.

Les risques sont classés en plusieurs catégories:

- Risque stratégique
- Risque opérationnel
- Risque financier
- Risque technique
- Risque externe

## Matrice des risques: Méthode

En face de chacun des risques qui a été identifié, il faudra en évaluer la gravité de l'impact ainsi que sa probabilité.

Suite à cela il sera possible de les organiser dans une matrice et traiter en priorité les risques qui auront la plus grande gravité et la plus forte probabilité.

## Matrice des risques: Exemple

	1 - Négligeable	2 - Mineure	3 - Modéré	4 - Majeure	5 - Catastrophique
5 - Très probable	5 - Membre de l'équipe malade			20 - Concurrent plus rapide	
4 - Probable	4 - Perte d'un disque dur			16 - Attaque sur la base de données	
3 - Possible					
2 - Peu Probable			6 - Coupure Internet		
1 - Très peu probable					5 - Coupure réseau totale

## Rédaction d'un plan de contingence

Le plan de contingence est un plan d'urgence qui décrit les actions à entreprendre en cas de survenance de risques.

Dans ce plan seront décrits les actions à effectuer en cas de survenue des risques ainsi que d'identifier les acteurs à prévenir ou qui auront des tâches à effectuer.

Ce plan doit être régulièrement mis à jour au fur et à mesure que de nouveaux risques sont identifiés ou que de nouveaux acteurs prennent part au projet.

## Le cahier des charges

Le cahier des charges est un document décrivant les objectifs du projet, les attentes ainsi que comment les mesurer et les livrables attendus pour le projet.

## Cahier des charges: Introduction

- Décrire brièvement l'objectif du projet.
- Fournir le contexte du projet, y compris les raisons pour lesquelles il est entrepris.
- Identifier les parties prenantes clés.

## Cahier des charges: Description du Projet

- Détailler les objectifs spécifiques du projet.
- Définir la portée du projet, y compris ce qui est inclus et ce qui est exclu.
- Lister les livrables attendus.

## Cahier des charges: Exigences Fonctionnelles

- Décrire les fonctionnalités attendues du produit ou du service.
- Définir les critères de performance attendus.

## Cahier des charges: Exigences Non Fonctionnelles

- Décrire les exigences de sécurité.
- Définir les exigences de fiabilité.
- Décrire les exigences d'utilisabilité.
- Définir les exigences de maintenabilité.

## Cahier des charges: Contraintes

- Décrire les contraintes techniques.
- Définir les contraintes budgétaires.
- Définir les contraintes temporelles.

## Cahier des charges: Gestion de la Qualité

- Lister les normes et réglementations applicables (par exemple, ISO 9001).
- Décrire les processus de contrôle qualité qui seront mis en place.

## Cahier des charges: Plan de Projet

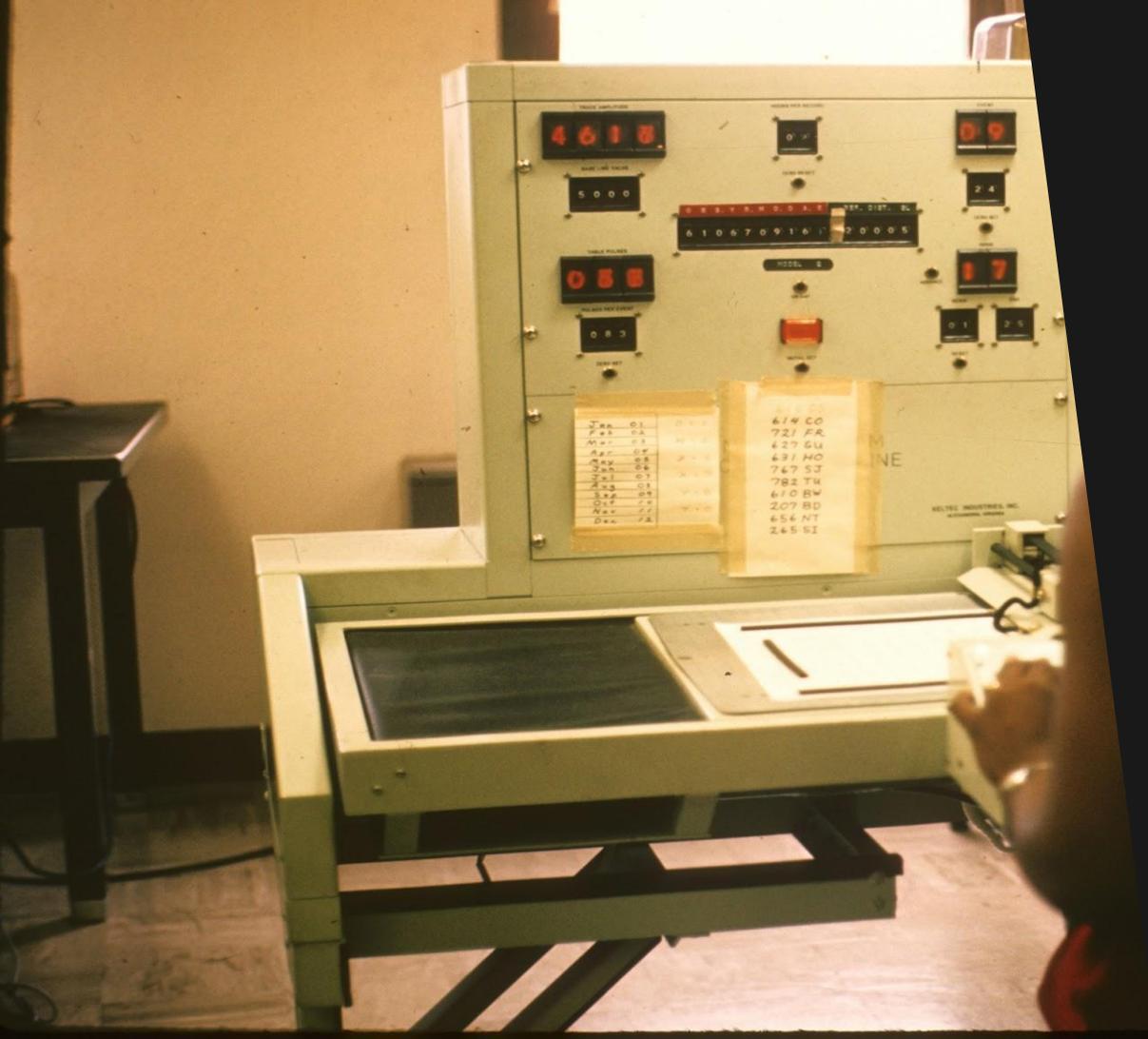
- Décrire les différentes phases du projet.
- Définir les jalons clés.
- Identifier les ressources nécessaires.

## **Cahier des charges: Acceptation**

- Définir les critères d'acceptation pour les livrables.
- Décrire la procédure d'acceptation.

# Plan du cours

- Introduction aux Concepts de Base de l'Organisation des Entreprises
- Cycle de vie et cycle de développement logiciel
- Activités de développement logiciel
- Avant-projet
- **Collecte des besoins : cas d'utilisation, user's story**
- Analyse Conception
- Codage
- Intégration
- Déploiement et production : plateforme de déploiement continu, surveillance des applications, conteneurisation en production (docker)
- Activités de gestion de projets



# Collecte des besoins

Cas d'utilisation, users story

## Cas d'utilisation

Un cas d'utilisation est l'ensemble des manières d'utiliser un système pour atteindre un but spécifique pour un utilisateur particulier. L'ensemble de tous les cas d'utilisation indique toutes les façons utiles d'utiliser un système.

## Cas d'utilisation: Contenu

- Désignation du cas d'utilisation : Est généralement un verbe, une action.
- Description brève.
- L'évènement déclencheur.
- L'enchaînement des événements du point de vue de l'utilisateur
  - On y indique les variantes
  - On peut y détailler les événements

## Cas d'utilisation: Contenu

- Les exigences particulières, comme les contraintes.
- Les préconditions, les conditions requises pour que le cas soit applicable.
- Les post-conditions, les résultat de l'application.
- Des extensions du cas d'utilisation initial.
- On peut y ajouter les acteurs, parties prenantes et des diagrammes dans les annexes.

## User story

Une user story est une description courte et simple d'une fonctionnalité du point de vue de l'utilisateur. Elle décrit ce que l'utilisateur veut faire et pourquoi il veut le faire.

Les user stories sont souvent utilisées dans les méthodologies agiles comme Scrum et Kanban.

## User Story: Définition

En tant que **<qui>**, je veux **<quoi>** afin de **<pourquoi>**

## User Story: Agile

Lorsque l'on utilise les user story dans les méthodes agiles, ces éléments servent généralement de backlog.

La complexité peut être chiffrée avec des points qu'on appelle des Story Points,

Idéalement une user story peut être développée en une itération.

## Persona

Lorsqu'une application dispose de plusieurs utilisateurs, il est possible de créer des Persona qui identifient un type d'utilisateur.

Ces persona doivent être créés à partir d'informations récoltées sur les utilisateurs actuels ou potentiels de l'application.

## Exemple de persona



Nom : Clownette Dupont

Âge : 28 ans

Sexe : Femme

Profession : Artiste de rue et animatrice de fêtes pour enfants

Localisation : Montmartre, Paris

## Exemple de persona: suite

Valeurs : Créativité, humour, joie de vivre.

Intérêts : Art de rue, théâtre, costumes, maquillage, jeux vidéo humoristiques.

Motivations : Faire rire les gens, exprimer sa créativité.

Frustrations : Applications trop sérieuses, interfaces ennuyeuses.

## Exemple de persona: suite

Habitudes d'Utilisation : Utilise son smartphone pour prendre des selfies et des vidéos amusantes, partage ses créations sur les réseaux sociaux.

Préférences : Applications avec des interfaces colorées et ludiques, effets spéciaux amusants.

Comportements d'Achat : Recherche des avis et des recommandations sur les réseaux sociaux, achète des applications qui promettent des expériences amusantes et créatives.

## Persona: Objectifs

Professionnels : Utiliser l'application pour créer des contenus amusants pour ses spectacles et ses animations, attirer plus de followers sur ses réseaux sociaux.

Personnels : Passer du bon temps avec ses amis en partageant des selfies amusants, exprimer sa créativité de manière ludique.

## Persona: Pain points

Problèmes Techniques : Applications qui plantent ou qui ont des bugs, fonctionnalités limitées.

Manque de Temps : Difficulté à trouver du temps pour créer des contenus amusants entre ses spectacles et ses animations.

Collaboration : Problèmes de communication avec ses amis pour organiser des défis humoristiques en ligne.

## Persona: Cas d'utilisation

Création de Selfies Amusants : Clownette utilise l'application pour transformer ses selfies en caricatures amusantes, ajoute des effets spéciaux et des filtres humoristiques.

Partage sur les Réseaux Sociaux : Clownette partage ses créations sur Instagram, TikTok et Facebook, et interagit avec ses followers en répondant aux commentaires et en participant à des défis.

Défis Humoristiques : Clownette organise des défis humoristiques avec ses amis, où chacun doit créer la caricature la plus drôle et la partager sur l'application.