Gestion des Plans de Tests - Bâtiment Intelligent (BMS/GTB)

Cours complet sur la gestion des plans de tests dans les systèmes de gestion technique de bâtiment (GTB/BMS), incluant la typologie des tests, la construction d'un plan de tests et les outils professionnels

Systèmes 45 min de lecture Niveau Intermédiaire

Document généré le 25/06/2026 à 21h38 · nouvy.fr/wiki/gestion-plans-tests-bms-gtb

Sommaire

36 section(s) · 45 min de lecture

Ressources à Télécharger

Introduction

1. Contexte du test dans les bâtiments intelligents

Un bâtiment intelligent (Smart Building) intègre :

- Des capteurs (température, CO_2, présence, luminosité…)
- Des automates (PLC, API industriels)
- Des systèmes de communication (BACnet, Modbus, KNX…)
- Des logiciels de supervision (BMS / GTB)
- Des scénarios et algorithmes d'optimisation énergétique

Chaque composant collecte, analyse ou traite des informations.

Toute erreur peut entraîner :

- Surconsommation d'énergie
- Panne d'équipement
- Inconfort utilisateur
- Perte financière
- Risque de sécurité

 D'où la nécessité d'un plan de tests structuré.

2. Définition d'un plan de tests

Un plan de tests est un document méthodologique listant tous les tests nécessaires pour valider un système, avec leurs conditions d'exécution, résultats attendus et critères de réussite.

Il garantit :

- La qualité du système
- La conformité au cahier des charges
- Le suivi du cycle de vie
- La traçabilité complète

Caractéristiques d'un bon plan de tests :

- **Exhaustif** : Couvre toutes les fonctionnalités critiques du système
- **Structuré** : Organisé par modules, sous-systèmes ou zones fonctionnelles
- **Traçable** : Chaque test peut être relié à une exigence du cahier des charges
- **Exécutable** : Les procédures sont claires et reproductibles
- **Mesurable** : Les critères de réussite sont quantifiables et objectifs

Avantages d'un plan de tests bien conçu :

- Réduction des risques de non-conformité
- Amélioration de la communication entre les équipes
- Facilitation de la maintenance future
- Documentation précieuse pour la formation des opérateurs
- Base solide pour les audits qualité

Partie I – La place du test dans le cycle de projet

1. Le cycle de vie d'un projet BMS

1. **Analyse des besoins**
    - Identification des exigences fonctionnelles
    - Définition des contraintes techniques
    - Établissement du cahier des charges
2. **Conception technique**
    - Architecture système
    - Choix des protocoles de communication
    - Sélection des équipements
3. **Installation matérielle & logicielle**
    - Pose des capteurs et actionneurs
    - Installation des automates
    - Déploiement des serveurs de supervision
4. **Configuration** (postes, automates, supervision)
    - Paramétrage des automates
    - Configuration des points de supervision
    - Création des scénarios d'automatisation
5. **Tests et validation**
    - Tests unitaires des composants
    - Tests d'intégration
    - Tests système globaux
    - Recette client
6. **Mise en production**
    - Mise en service progressive
    - Formation des utilisateurs
    - Documentation finale
7. **Maintenance & évolution**
    - Maintenance préventive
    - Corrections de bugs
    - Évolutions fonctionnelles

Les tests interviennent à plusieurs moments :

- **Avant livraison** : Validation complète du système installé

- **Avant exploitation** : Vérification finale avant mise en service
- **Après mise à jour** : Tests de non-régression après modifications
- **Lors de travaux d'extension** : Validation des nouveaux équipements intégrés
- **Maintenance préventive** : Tests périodiques pour garantir le bon fonctionnement

2. Trois niveaux de tests dans le BMS

A. Test composant « Unitaire »

Test d'un élément isolé, indépendamment du reste du système.

Éléments testés :

- **Fonctionnement d'un capteur** : Vérification de la précision, de la plage de mesure, de la linéarité
- **API d'automate** : Test des entrées/sorties numériques et analogiques
- **Actionneur** (vanne, moteur…) : Vérification de la course, du temps de réponse, de la position

Exemple concret :

Test d'un capteur de température :

- Placer le capteur dans un environnement à température connue (ex: 20°C)
- Vérifier que la valeur lue correspond à ±0,5°C
- Tester la réponse à un changement rapide de température
- Vérifier la stabilité de la mesure sur 24h

Avantages :

- Détection précoce des défauts
- Isolation des problèmes
- Tests rapides et ciblés

B. Test d'intégration

On valide l'interaction entre plusieurs composants du système.

Chaînes de communication testées :

- **Capteur → Automate** : Vérification de la remontée des valeurs
- **Automate → Serveur** : Test de la communication réseau (BACnet, Modbus, etc.)
- **Serveur → Supervision** : Validation de l'affichage et de la mise à jour des données
- **Supervision → Rapport / Dashboard** : Vérification de la génération des rapports

Exemple concret :

Test d'intégration capteur de CO_2 :

1. Le capteur mesure 1200 ppm de CO_2
2. L'automate reçoit la valeur via Modbus
3. Le serveur BMS récupère la donnée via BACnet
4. La supervision affiche l'alarme "CO_2 élevé"
5. Le système déclenche l'augmentation du débit de ventilation

Points critiques à vérifier :

- Temps de propagation de la donnée
- Perte de données éventuelle
- Format et unités des valeurs
- Gestion des erreurs de communication

C. Test système (global)

On simule un scénario réel complet impliquant plusieurs sous-systèmes.

Exemple détaillé : Gestion de la qualité de l'air

Scénario : "Si CO_2 > 900 ppm dans une salle occupée

- Alors augmenter le débit ventilation de 50%
- Et envoyer l'alarme en supervision
- Et enregistrer l'événement dans les logs"

Procédure de test :

1. **Préparation** :
    - Salle équipée de capteur CO_2, capteur présence, et ventilation variable
    - Supervision configurée avec alarmes CO_2
    - Système en mode automatique
2. **Exécution** :
    - Simuler présence dans la salle (capteur présence = ON)
    - Injecter du CO_2 pour atteindre 950 ppm
    - Observer le comportement du système
3. **Vérifications** :
    - □ Le capteur CO_2 détecte la valeur > 900 ppm
    - □ L'alarme apparaît en supervision dans les 2 secondes
    - □ Le débit de ventilation augmente de 50% en moins de 5 secondes
    - □ L'événement est enregistré dans les logs avec timestamp
    - □ L'alarme disparaît quand CO_2 < 800 ppm

Avantages des tests système :

- Validation du comportement réel du système
- Détection des problèmes d'interaction entre composants
- Vérification de la cohérence globale

Partie II – Typologie des tests

1. Tests fonctionnels

Objectif : vérifier que chaque fonction fait exactement ce qu'elle doit faire selon les spécifications.

Principe : Pour chaque fonctionnalité, on définit :

- Les conditions d'entrée
- Le comportement attendu
- Les critères de réussite

Exemples détaillés :

Exemple 1 : Ouverture d'une vanne

Fonction : Vanne motorisée doit s'ouvrir à 50% sur commande

Test :

- **Précondition** : Vanne fermée (0%)
- **Action** : Envoyer commande "Ouvrir à 50%"
- **Résultat attendu** :
    - Vanne atteint 50% en moins de 10 secondes
    - Position confirmée par retour de mesure
    - Pas de sur-oscillation (> 55% ou < 45%)

Exemple 2 : Régulation chauffage

Fonction : Maintenir la température à 20°C ± 0,5°C

Test :

- **Précondition** : Température ambiante = 18°C
- **Action** : Activer la régulation avec consigne 20°C
- **Résultat attendu** :
    - Température atteint 20°C en moins de 30 minutes
    - Température stabilisée entre 19,5°C et 20,5°C
    - Pas de surchauffe (> 21°C)

Exemple 3 : Alerte défaut automate

Fonction : Détecter et remonter une panne d'automate

Test :

- **Précondition** : Automate en fonctionnement normal
- **Action** : Débrancher l'alimentation de l'automate
- **Résultat attendu** :
    - Alarme "Automate déconnecté" apparaît en supervision dans les 5 secondes
    - Alarme de niveau critique (rouge)
    - Notification envoyée à l'opérateur

Exemple 4 : Reporting de consommation

Fonction : Générer un rapport quotidien de consommation énergétique

Test :

- **Précondition** : Système en fonctionnement depuis 24h avec données de consommation
- **Action** : Générer le rapport du jour précédent
- **Résultat attendu** :
    - Rapport généré en moins de 30 secondes
    - Toutes les données présentes (consommation totale, par zone, par équipement)
    - Format PDF lisible et bien formaté
    - Envoi automatique par email si configuré

Critères principaux de validation :

- **Déclenchement** : La fonction se déclenche-t-elle au bon moment ?
- **Cohérence des valeurs** : Les valeurs sont-elles logiques et dans les plages attendues ?
- **Temps de réponse** : Le système réagit-il dans les délais spécifiés ?
- **Robustesse** : Le système gère-t-il correctement les cas limites (valeurs extrêmes, pannes) ?
- **Précision** : Les mesures et actions sont-elles suffisamment précises ?

2. Tests d'intégration

Objectif : valider la communication et l'interaction entre différents éléments du système.

Enjeux : Les systèmes BMS utilisent souvent plusieurs protocoles de communication différents. Il est essentiel de vérifier que tous les équipements communiquent correctement entre eux.

Vérifications typiques détaillées :

A. Tests de protocoles de communication

BACnet :

- Vérifier que les objets BACnet sont correctement découverts
- Tester la lecture/écriture des propriétés (Present Value, Status Flags)
- Valider les trames BACnet avec un analyseur réseau
- Vérifier la gestion des erreurs (timeout, device non disponible)

Modbus :

- Tester la lecture des registres (coils, holding registers, input registers)
- Vérifier l'écriture des commandes
- Valider la gestion des adresses esclaves
- Tester la communication sur différents baud rates

KNX :

- Vérifier que les objets de groupe sont accessibles
- Tester les télégrammes KNX (read, write, response)
- Valider la configuration des adresses physiques et de groupe

IP/MSTP :

- Tester la connectivité réseau (ping, traceroute)
- Vérifier la configuration des adresses IP
- Valider le routage MSTP (Multi-Stage Token Passing) pour BACnet
- Tester la résistance aux pannes réseau (coupure de câble)

B. Tests de remontée de données

Scénario complet :

1. **Capteur → Automate** :
    - Capteur envoie une valeur (ex: température 22°C)
    - Automate reçoit et traite la valeur
    - Vérifier la latence (< 1 seconde généralement)
2. **Automate → Serveur** :
    - Serveur interroge l'automate via le protocole
    - Automate répond avec la valeur
    - Vérifier que la valeur n'est pas altérée
3. **Serveur → Supervision** :
    - Supervision récupère la donnée du serveur
    - Affichage en temps réel
    - Vérifier la mise à jour automatique

Points critiques :

- **Perte de données** : Aucune donnée ne doit être perdue lors de la transmission
- **Synchronisation** : Les valeurs doivent être cohérentes entre les différents niveaux
- **Gestion des erreurs** : Le système doit gérer correctement les pannes de communication
- **Performance** : Le système doit supporter le nombre de points configurés sans dégradation

C. Tests de résilience

Scénarios de panne :

- Coupure réseau entre automate et serveur
- Automate redémarré pendant la communication
- Surcharge réseau (trop de données simultanées)
- Interférences électromagnétiques

Comportement attendu :

- Reconnexion automatique après rétablissement

- Mise en cache des données pendant la panne
- Synchronisation automatique après reconnexion
- Pas de perte de données critiques

3. Tests de performance

Objectif : Vérifier que le système répond aux exigences de performance définies dans le cahier des charges.

Les tests de performance permettent de s'assurer que le système peut gérer la charge prévue et répondre dans les temps requis.

Paramètres mesurés :

Paramètre	Description	Exemple	Critère typique
Latence	Temps entre un événement et la réaction du système	Temps entre détection capteur et ordre automate	< 2 secondes
Charge	Nombre de données traitées par unité de temps	Nombre max de données par minute	1000 points/min
Scalabilité	Capacité à fonctionner avec différents volumes	Fonctionne-t-il avec 10 objets ? Avec 1000 ?	Jusqu'à 5000 points
Disponibilité	Pourcentage de temps de fonctionnement	Taux de disponibilité sur 30 jours	> 99,5%
Débit réseau	Quantité de données transmises	Mo/s sur le réseau BACnet	Selon infrastructure
Temps de réponse	Délai de réponse aux commandes	Temps pour ouvrir une vanne	< 5 secondes

Méthodologie de test :

A. Test de latence

Procédure :

1. Enregistrer l'heure T1 lors d'un changement de valeur capteur
2. Mesurer l'heure T2 quand la commande arrive à l'actionneur
3. Calculer : Latence = T2 - T1

Exemple concret :

- Capteur de température détecte un changement à 10:00:00.000
- Commande arrive au chauffage à 10:00:01.500
- Latence = 1,5 secondes (critère : < 2 secondes)

B. Test de charge

Procédure :

1. Simuler un grand nombre de points (capteurs, actionneurs)

2. Générer des changements de valeurs simultanés
3. Mesurer le temps de traitement et la perte éventuelle de données

Scénario :

- 500 capteurs qui changent de valeur simultanément
- Système doit traiter toutes les données en moins de 10 secondes
- Aucune donnée ne doit être perdue

C. Test de scalabilité

Procédure progressive :

1. Tester avec 10 points → OK
2. Tester avec 100 points → OK
3. Tester avec 1000 points → OK
4. Tester avec 5000 points → Vérifier les limites

Indicateurs à surveiller :

- Temps de réponse qui augmente
- Consommation mémoire
- Charge CPU
- Saturation réseau

D. Test de disponibilité

Méthode :

- Monitoring continu pendant 30 jours
- Enregistrement de toutes les pannes
- Calcul : Disponibilité = (Temps total - Temps de panne) / Temps total × 100

Exemple :

- Période : 30 jours = 43 200 minutes
- Pannes totales : 100 minutes
- Disponibilité = (43 200 - 100) / 43 200 × 100 = 99,77% □

Outils utilisés :

- **Analyseurs réseau** : Wireshark, tcpdump pour analyser le trafic
- **Scripts de charge** : Génération automatique de données de test
- **Monitoring GTB** : Outils intégrés de supervision pour mesurer les performances
- **Oscilloscopes/logiciels** : Pour mesurer les temps de réponse précis
- **Outils de benchmarking** : Pour comparer les performances avec des références

4. Tests de sécurité

Objectif : Vérifier que le système est protégé contre les accès non autorisés et les attaques malveillantes.

Les systèmes BMS sont de plus en plus connectés aux réseaux d'entreprise et à Internet, ce qui les expose à des risques de sécurité. Les tests de sécurité sont essentiels pour protéger

les bâtiments et leurs occupants.

Contrôles détaillés :

A. Authentification et autorisation

Mots de passe :

- □ Tous les mots de passe par défaut ont été changés
- □ Politique de mots de passe forte (min 12 caractères, majuscules, chiffres, caractères spéciaux)
- □ Rotation périodique des mots de passe
- □ Pas de mots de passe en clair dans les fichiers de configuration

Gestion des utilisateurs :

- □ Chaque utilisateur a un compte unique (pas de compte partagé)
- □ Principe du moindre privilège (droits minimaux nécessaires)
- □ Désactivation automatique des comptes inactifs
- □ Audit des connexions utilisateurs

B. Sécurité réseau

Chiffrement des communications :

- □ Utilisation de protocoles sécurisés (HTTPS, TLS pour BACnet/IP)
- □ Certificats SSL valides et à jour
- □ Pas de communication en clair pour les données sensibles

Cloisonnement réseau :

- □ Séparation des réseaux BMS et réseau d'entreprise (VLAN)
- □ Pare-feu configuré entre les réseaux
- □ Règles de filtrage des ports et protocoles
- □ Isolation des équipements critiques

Sécurisation des automates :

- □ Désactivation des services inutiles (Telnet, FTP)
- □ Mise à jour des firmwares (correction des vulnérabilités)
- □ Configuration des règles de pare-feu sur les automates

C. Journalisation et audit

Journalisation des accès :

- □ Enregistrement de toutes les connexions (succès et échecs)
- □ Traçabilité des actions critiques (changement de consigne, arrêt d'équipement)
- □ Conservation des logs pendant au moins 1 an
- □ Protection des logs contre la modification

Audit de sécurité :

- □ Revue régulière des logs d'accès
- □ Détection des tentatives d'intrusion

- □ Alertes automatiques en cas d'activité suspecte

D. Tests de pénétration

Objectif : Simuler des attaques pour identifier les vulnérabilités.

Types de tests :

1. **Scan de ports** : Identifier les ports ouverts et services accessibles
2. **Test de force brute** : Tentative de connexion avec des mots de passe courants
3. **Injection de commandes** : Tentative d'injection dans les protocoles (Modbus, BACnet)
4. **Test de déni de service** : Vérifier la résistance aux surcharges

Les attaques BMS courantes :

Type d'attaque	Description	Impact	Protection
Prise de contrôle automates	Accès non autorisé aux automates	Contrôle total du système	Authentification forte, réseau isolé
Injection Modbus	Envoi de commandes malveillantes via Modbus	Manipulation des équipements	Validation des commandes, liste blanche
Montée d'inconfort (attaque "soft")	Modification subtile des consignes	Inconfort des occupants	Détection des anomalies, alertes
Arrêt technique (attaque "hard")	Arrêt brutal des équipements	Panne totale du système	Redondance, isolation réseau
Vol de données	Accès aux données de consommation	Espionnage industriel	Chiffrement, contrôle d'accès

Exemple de test de sécurité :

Scénario : Test d'accès non autorisé à un automate

1. **Préparation** : Automate configuré avec authentification
2. **Test** : Tentative de connexion avec mot de passe par défaut
3. **Résultat attendu** : Accès refusé, tentative enregistrée dans les logs
4. **Vérification** : Alerte envoyée à l'administrateur

Checklist de sécurité BMS :

- □ Tous les mots de passe par défaut changés
- □ Chiffrement activé sur toutes les communications
- □ Réseau BMS isolé du réseau d'entreprise
- □ Pare-feu configuré et actif
- □ Mises à jour de sécurité appliquées
- □ Journalisation activée et vérifiée
- □ Tests de pénétration réalisés
- □ Plan de réponse aux incidents défini

5. Tests utilisateurs (UAT - User Acceptance Testing)

Objectif : Valider que le système répond aux besoins réels des utilisateurs finaux et qu'il est utilisable en conditions d'exploitation.

Les tests UAT sont réalisés par les futurs utilisateurs du système, avec des scénarios réels d'utilisation. C'est la dernière étape avant la mise en production.

Participants :

- Utilisateurs finaux (exploitants, techniciens de maintenance)
- Représentants du client
- Équipe projet (support technique)

Ce que le client vérifie :

A. Utilisabilité de l'interface

Critères d'évaluation :

- **Navigation intuitive** : Les utilisateurs trouvent facilement les informations
- **Lisibilité** : Les écrans sont clairs et bien organisés
- **Réactivité** : L'interface répond rapidement aux actions
- **Ergonomie** : Les actions courantes sont accessibles rapidement

Exemples de tests :

- Un exploitant doit pouvoir consulter la température d'une salle en moins de 3 clics
- Les alarmes doivent être visibles immédiatement
- Les graphiques doivent être lisibles et compréhensibles

B. Adaptation des scénarios

Vérification :

- Les scénarios automatiques correspondent aux besoins réels
- Les plages horaires sont correctes (heures d'ouverture, jours fériés)
- Les consignes sont adaptées aux usages (température, éclairage)
- Les priorités sont respectées (ex: confort vs économie d'énergie)

Exemple :

- Scénario "Mode nuit" : Vérifier que l'éclairage s'éteint bien après la fermeture
- Scénario "Présence" : Vérifier que la ventilation s'active correctement quand quelqu'un entre dans la salle

C. Gestion des droits utilisateurs

Vérification des rôles :

- **Administrateur** : Accès complet à la configuration
- **Exploitant** : Accès à la supervision et aux commandes manuelles
- **Lecteur** : Accès en lecture seule aux données

Tests de sécurité :

- Un utilisateur "Lecteur" ne peut pas modifier les consignes
- Un utilisateur "Exploitant" ne peut pas modifier la configuration système
- Les tentatives d'accès non autorisé sont bloquées

D. Formation et documentation

Vérification :

- La documentation utilisateur est complète et claire
- Les utilisateurs peuvent utiliser le système après formation
- Les procédures d'exploitation sont documentées
- Les procédures de maintenance sont disponibles

Dans l'industrie, cela s'appelle souvent :

- **"Recette Client"** : Validation finale par le client avant acceptation
- **"Mise en service"** : Passage du système en exploitation
- **"Livraison"** : Remise du système au client

Procédure de recette client :

1. **Préparation** :
    - Système entièrement testé et validé par l'équipe projet
    - Documentation complète disponible
    - Formation des utilisateurs réalisée
2. **Exécution** :
    - Tests réalisés par les utilisateurs avec scénarios réels
    - Validation de chaque fonctionnalité
    - Identification des éventuels points d'amélioration
3. **Validation** :
    - Signature du procès-verbal de recette
    - Liste des réserves (points à corriger)
    - Planification des corrections si nécessaire

Résultat attendu :

- Système conforme aux attentes du client
- Utilisateurs formés et autonomes
- Documentation complète et validée
- Système prêt pour l'exploitation

Partie III – Construction d'un plan de tests

1. Structure complète professionnelle

Un plan de tests complet contient :

A. Contexte et périmètre

Définition du système :

- Description détaillée du système à tester
- Liste des équipements concernés
- Architecture système (schéma fonctionnel)
- Version des logiciels et firmwares

Limites du test :

- Ce qui est inclus dans les tests
- Ce qui est exclu (hors périmètre)
- Contraintes techniques ou temporelles

Hypothèses :

- Environnement de test disponible (banc d'essai + simulateur)
- Accès aux équipements nécessaires
- Documentation technique disponible
- Équipe formée aux outils de test

Exemple :

> ***Système** : Gestion technique de bâtiment pour un immeuble de bureaux de 5000 m²*
> ***Périmètre inclus** :*
>
> - Système de chauffage/ventilation/climatisation (CVC)
> - Éclairage intelligent
> - Gestion des accès
>
> ***Périmètre exclu** :*
>
> - Système d'ascenseurs (géré par un autre prestataire)
> - Système de sécurité incendie (norme spécifique)
>
> ***Hypothèses** :*
>
> - Banc d'essai disponible avec simulateurs de capteurs
> - Accès au réseau de test isolé
> - Documentation BACnet et Modbus disponible

B. Références

Documents de référence :

- **Cahier des charges** : Spécifications fonctionnelles et techniques
- **Carnet d'intégration** : Documentation technique des équipements
- **Normes techniques** : Normes applicables (ISO, EN, etc.)
- **Manuels utilisateur** : Documentation des équipements
- **Plans d'architecture** : Schémas électriques, réseaux, etc.

Exemple de références :

- Cahier des charges v2.3 du 15/03/2024
- Norme EN ISO 16484 (BACnet)
- Manuel technique Schneider EcoStruxure v4.2
- Plan réseau GTB - Version finale

C. Organisation

Équipe projet :

- **Responsable test** : Personne en charge de l'exécution des tests
- **Responsable validation** : Personne validant les résultats
- **Testeurs** : Équipe exécutant les tests
- **Support technique** : Assistance en cas de problème

Calendrier prévu :

- Date de début des tests
- Durée estimée de chaque phase
- Jalons de validation
- Date de fin prévue

Exemple de planning :

Phase	Durée	Responsable
Tests unitaires	1 semaine	Équipe intégration
Tests d'intégration	2 semaines	Équipe test
Tests système	1 semaine	Équipe test + Client
Recette client	3 jours	Client

D. Environnement de test

Composants de l'environnement :

- **Supervision de préproduction** : Copie de la supervision de production pour tests
- **Simulateur de capteurs** : Outil permettant de simuler les valeurs des capteurs
- **Réseau isolé** : Réseau dédié pour éviter d'impacter la production
- **Automates réels** : Automates de test ou copie de la configuration production
- **Serveurs de test** : Serveurs BMS dédiés aux tests

Configuration type :

```
Environnement de test :
├── Serveur BMS (préproduction)
├── Supervision (copie production)
├── 3 automates de test
├── Simulateur de capteurs (20 points)
├── Réseau isolé (VLAN test)
└── Poste de travail testeur
```

Copier

Avantages d'un environnement dédié :

- Pas d'impact sur la production
- Tests reproductibles
- Possibilité de simuler des pannes
- Tests de charge sans risque

E. Liste des cas de tests

Chaque cas de test doit inclure :

Champ	Description
ID	Numéro unique
Priorité	Haute / Moyenne / Basse
Objet testé	Capteur / API / Supervision
Objectif	But du test
Préconditions	Ce qu'il faut avant de commencer
Procédure	Étapes détaillées
Résultat attendu	Valeur exacte de réussite
Résultat obtenu	Valeur mesurée
Statut	OK / NOK
Date	D'exécution
Testeur	Opérateur

2. Exemples professionnalisés de cas de test

Exemple 1 : TC-418 – Test de régulation chauffage

Élément	Valeur
ID	TC-418
Priorité	Haute
Type	Test fonctionnel
Sous-système	CVC - Chauffage
Objet testé	Régulation température salle 201
Objectif	Vérifier que le chauffage s'active automatiquement si la température est inférieure à la consigne
Préconditions	- Capteur de température installé et calibré- Automate de régulation configuré- Consigne réglée à 19°C- Système en mode automatique
Procédure	1. Vérifier l'état initial : température = 20°C, chauffage OFF2. Simuler une baisse de température à 18°C via simulateur3. Observer l'automate de régulation4. Vérifier que la commande "Chauffage ON" est envoyée5. Mesurer le temps de réaction
Résultat attendu	- Chauffage activé en moins de 4 secondes- Température remonte progressivement- Arrêt automatique quand T° ≥ 19°C
Résultat obtenu	(à remplir lors de l'exécution)
Statut	OK / NOK / Bloqué
Date d'exécution	(à remplir)
Testeur	(nom du testeur)
Commentaires	

Exemple 2 : TC-502 – Test d'intégration capteur CO_2

Élément	Valeur
ID	TC-502
Priorité	Haute
Type	Test d'intégration
Sous-système	Qualité de l'air
Objet testé	Chaîne complète : Capteur CO_2 → Automate → Supervision
Objectif	Vérifier que la valeur CO_2 remonte correctement de la source jusqu'à la supervision
Préconditions	- Capteur CO_2 connecté à l'automate- Automate configuré en Modbus- Serveur BMS connecté- Supervision configurée avec le point CO_2
Procédure	1. Injecter une valeur connue au capteur (ex: 800 ppm)2. Vérifier la valeur dans l'automate (lecture Modbus)3. Vérifier la valeur dans le serveur BMS4. Vérifier l'affichage en supervision5. Comparer les valeurs à chaque niveau
Résultat attendu	- Valeur identique à tous les niveaux (800 ppm ± 10 ppm)- Latence totale < 2 secondes- Pas de perte de données
Résultat obtenu	(à remplir lors de l'exécution)
Statut	OK / NOK / Bloqué
Date d'exécution	(à remplir)
Testeur	(nom du testeur)
Commentaires	

Exemple 3 : TC-715 – Test de sécurité authentification

Élément	Valeur
ID	TC-715
Priorité	Critique
Type	Test de sécurité
Sous-système	Authentification
Objet testé	Système d'authentification supervision
Objectif	Vérifier que les tentatives d'accès non autorisé sont bloquées
Préconditions	- Compte utilisateur "test" créé avec mot de passe "Test123!"- Supervision accessible- Journalisation activée
Procédure	1. Tentative de connexion avec mauvais mot de passe (3 tentatives)2. Vérifier le blocage du compte3. Vérifier l'enregistrement dans les logs4. Vérifier l'envoi d'alerte à l'administrateur
Résultat attendu	- Accès refusé après 3 tentatives- Compte bloqué pendant 15 minutes- Tentatives enregistrées dans les logs- Alerte envoyée à l'admin
Résultat obtenu	(à remplir lors de l'exécution)
Statut	OK / NOK / Bloqué
Date d'exécution	(à remplir)
Testeur	(nom du testeur)
Commentaires	

3. Expression des résultats

A. Tableaux de synthèse

Tableau récapitulatif global :

Indicateur	Valeur	Pourcentage
Nombre total de tests	150	100%
Tests réussis (OK)	142	94,6%
Tests en échec (NOK)	6	4,0%
Tests bloqués	2	1,4%
Taux de succès	-	94,6%

Répartition par type de test :

Type de test	Total	OK	NOK	Taux succès
Tests fonctionnels	80	76	4	95,0%
Tests d'intégration	40	38	2	95,0%
Tests de performance	15	15	0	100%
Tests de sécurité	10	9	1	90,0%
Tests UAT	5	4	0	80,0%

Répartition par priorité :

Priorité	Total	OK	NOK	Taux succès
Critique	20	19	1	95,0%
Haute	60	57	3	95,0%
Moyenne	50	48	2	96,0%
Basse	20	18	0	90,0%

B. Indicateurs complémentaires

Qualité :

- **Taux d'anomalies critiques** : Nombre d'anomalies critiques / Nombre total de tests
    - Exemple : 2 anomalies critiques sur 150 tests = 1,3%
- **Taux de non-régressions réussies** : Tests de non-régression OK / Total tests de non-régression
    - Exemple : 45/50 = 90%

- **Couverture fonctionnelle** : Pourcentage de fonctionnalités testées
    - Exemple : 95% des fonctionnalités critiques testées

Performance :

- **Temps moyen de résolution** : Temps moyen pour corriger une anomalie
    - Exemple : 2,5 jours en moyenne
- **Taux de réouverture** : Nombre d'anomalies réouvertes / Nombre d'anomalies corrigées
    - Exemple : 2/20 = 10%

Efficacité :

- **Taux d'exécution** : Tests exécutés / Tests planifiés
    - Exemple : 150/160 = 93,75%
- **Productivité** : Nombre de tests exécutés par jour
    - Exemple : 15 tests/jour

C. Graphiques et visualisations

Graphique en camembert : Répartition des résultats (OK/NOK/Bloqué)

Graphique en barres : Résultats par type de test

Courbe d'évolution : Progression du taux de succès au fil du temps

Carte de chaleur : Répartition des anomalies par sous-système

D. Analyse des résultats

Points positifs :

- □ Taux de succès global élevé (94,6%)
- □ Tous les tests critiques réussis sauf 1
- □ Tests de performance à 100%
- □ Bonne couverture fonctionnelle

Points d'attention :

- ⚠ 2 tests bloqués nécessitent une investigation
- ⚠ Tests UAT à améliorer (80% de succès)
- ⚠ 1 anomalie critique à corriger avant mise en production

Recommandations :

- Corriger l'anomalie critique identifiée
- Réexécuter les tests bloqués après correction
- Améliorer les tests UAT avec le client
- Valider les corrections avant mise en production

Partie IV – Gestion des anomalies

1. Cycle de vie d'une anomalie

Le cycle de vie d'une anomalie suit un processus structuré pour garantir sa résolution complète.

Étapes détaillées :

1. **Détection**
    - Identification de l'anomalie lors d'un test
    - Enregistrement dans le système de suivi
    - Attribution d'un identifiant unique (ex: BUG-117)
2. **Qualification**
    - Description détaillée du problème
    - Détermination de la sévérité (Critique / Majeure / Mineure)
    - Évaluation de la reproductibilité
    - Capture de preuves (screenshots, logs, valeurs)
3. **Affectation**
    - Attribution à un responsable (développeur / intégrateur / fournisseur)
    - Définition de la priorité selon la sévérité
    - Estimation du temps de correction
4. **Correction**
    - Analyse de la cause racine
    - Développement de la correction
    - Tests unitaires de la correction
    - Mise à disposition en environnement de test
5. **Re-test**
    - Réexécution du test qui avait échoué
    - Vérification que la correction résout le problème
    - Tests de non-régression (vérifier qu'on n'a pas cassé autre chose)
6. **Clôture**
    - Validation de la correction
    - Mise à jour de la documentation si nécessaire
    - Fermeture de l'anomalie dans le système de suivi

Durée typique du cycle :

Sévérité	Durée cible	Exemple
Critique	24-48h	Panne système
Majeure	3-5 jours	Fonctionnalité importante
Mineure	1-2 semaines	Amélioration cosmétique

2. Fiche anomalie – Modèle détaillé

Modèle complet de fiche d'anomalie :

Champ	Description	Exemple
ID	Identifiant unique de l'anomalie	BUG-117
Titre	Résumé court du problème	Vanne ne réagit pas en mode AUTO
Sévérité	Impact sur le système	Critique / Majeure / Mineure
Priorité	Urgence de traitement	P0 (immédiat) / P1 / P2 / P3
Type	Catégorie de l'anomalie	Fonctionnel / Intégration / Performance / Sécurité
Sous-système	Zone concernée	CVC - Ventilation - Local 2
Description	Description détaillée du problème	La vanne V-201 ne répond pas aux commandes en mode automatique. En mode manuel, la vanne fonctionne correctement.
Environnement	Contexte de l'anomalie	Réseau BACnet – Local 2 – Automate A-05
Reproductibilité	Fréquence d'apparition	100% (toujours) / 50% (parfois) / Rare
Étapes pour reproduire	Procédure pour reproduire le problème	1. Mettre la vanne en mode AUTO2. Envoyer commande "Ouvrir à 50%"3. Observer : pas de mouvement4. Passer en mode MANUEL5. Commande fonctionne
Résultat attendu	Comportement normal attendu	Vanne s'ouvre à 50% en moins de 10 secondes
Résultat observé	Comportement réel observé	Pas de mouvement, pas d'erreur affichée
Fichiers joints	Captures, logs, etc.	Screenshot supervision, log automate
Responsable	Personne en charge de la correction	Intégrateur GTB - Jean Dupont
Date création	Date de détection	15/03/2024
Date résolution	Date de correction	(à remplir)
État	Statut actuel	Ouvert / En cours / Corrigé / Testé / Clos / Rejeté
Commentaires	Notes additionnelles	Problème identifié : configuration automate incorrecte

Exemple complet :

***BUG-117** - Vanne V-201 ne répond pas en mode AUTO*
***Sévérité** : Majeure | **Priorité** : P1 | **Type** : Fonctionnel*
***Description** : La vanne motorisée V-201 du local 2 ne répond pas aux commandes automatiques. Le mode manuel fonctionne correctement, ce qui indique un problème de configuration plutôt qu'un problème matériel.*
***Environnement** : Réseau BACnet, Automate A-05, Supervision EcoStruxure v4.2*
***Reproductibilité** : 100% - Le problème se produit systématiquement*
***Étapes pour reproduire** :*

1. Se connecter à la supervision
2. Sélectionner la vanne V-201
3. Passer en mode AUTO
4. Envoyer commande "Ouvrir à 50%"
5. Observer : aucun mouvement de la vanne
6. Vérifier les logs de l'automate : aucune commande reçue

***Résultat attendu** : Vanne s'ouvre à 50% en moins de 10 secondes*
***Résultat observé** : Pas de mouvement, pas d'erreur dans les logs*
***Fichiers joints** :*

- Screenshot supervision (bug-117-screenshot.png)
- Log automate A-05 (bug-117-logs.txt)

***Responsable** : Intégrateur GTB - Jean Dupont*
***État** : En cours*
***Commentaires** :*

- 16/03/2024 : Analyse en cours, vérification de la configuration BACnet
- 17/03/2024 : Problème identifié : objet BACnet mal configuré dans l'automate

---

Partie V – Outils professionnels

1. Outils logiciels GTB utilisés en industrie

Supervision et configuration :

- **Schneider EcoStruxure** : Plateforme de gestion de bâtiment connecté
- **Siemens Desigo** : Système de gestion technique de bâtiment
- **Niagara Framework** : Plateforme ouverte pour l'intégration de systèmes
- **Tridium AX** : Framework de développement pour systèmes intégrés
- **WIT** : Logiciel de supervision et de gestion énergétique
- **Omnicore** : Solution de gestion technique de bâtiment

Fonctionnalités communes :

- Configuration des automates
- Supervision en temps réel
- Historisation des données
- Génération de rapports

- Gestion des alarmes

2. Outils réseau et protocoles

Analyse de trames réseau :

- **Wireshark** : Analyseur de protocoles réseau, permet de capturer et analyser les trames BACnet, Modbus, etc.
- **tcpdump** : Outil en ligne de commande pour l'analyse réseau

Exploration de protocoles BMS :

- **BACnet Explorer** : Outil dédié pour explorer et tester les objets BACnet
- **Modbus Doctor** : Logiciel pour tester et déboguer les communications Modbus
- **KNX ETS** : Outil de configuration et test pour les installations KNX

Tests et automatisation :

- **TestLink** : Gestion de campagnes de tests
- **Squash** : Plateforme de test et qualité logicielle
- **JMeter** : Tests de charge et performance
- **Postman** : Tests d'API REST pour les systèmes BMS modernes

Utilisation typique :

1. **Wireshark** : Capturer les échanges réseau pour diagnostiquer un problème de communication
2. **BACnet Explorer** : Découvrir les objets BACnet disponibles sur le réseau
3. **Modbus Doctor** : Tester la lecture/écriture de registres Modbus
4. **Postman** : Tester les API REST d'un système BMS moderne

3. Outils physiques de mesure

Mesures environnementales :

- **Luxmètre** : Mesure de l'éclairement (lux) pour valider les capteurs de luminosité
- **Hygromètre** : Mesure de l'humidité relative pour calibrer les capteurs
- **Thermomètre / Thermocouple** : Mesure de température de référence pour étalonnage
- **Anémomètre** : Mesure de la vitesse de l'air pour les systèmes de ventilation
- **Débitmètre** : Mesure du débit d'air ou d'eau pour les systèmes CVC

Mesures électriques :

- **Multimètre** : Mesure de tension, courant, résistance pour diagnostiquer les équipements
- **Pince ampèremétrique** : Mesure de courant sans déconnexion
- **Oscilloscope** : Visualisation des signaux électriques pour l'analyse fine

Analyse réseau :

- **Analyseurs IP/MSTP** : Outils spécialisés pour analyser les réseaux BACnet
- **Testeurs de câbles** : Vérification de l'intégrité des câbles réseau
- **Toner de réseau** : Localisation et identification des câbles

Diagnostic avancé :

- **Caméra thermique** : Détection des surchauffes et problèmes d'isolation
- **Analyseur de qualité d'air** : Mesure CO_2, COV, particules pour validation des capteurs
- **Enregistreur de données** : Enregistrement continu de plusieurs paramètres

Exemple d'utilisation :

Pour valider un capteur de température :

1. Placer un thermomètre de référence à côté du capteur
2. Comparer les valeurs affichées
3. Vérifier la précision (écart < ±0,5°C généralement)
4. Tester sur toute la plage de mesure

Partie VI – Vocabulaire essentiel

Terme	Définition
Test	Vérification du bon fonctionnement d'un composant ou d'un processus
Plan de tests	Document qui décrit l'ensemble des tests à réaliser
Cas de test (Test Case)	Unité de test décrivant : objectif, prérequis, action, résultat attendu
Campagne de test	Ensemble de cas de tests exécutés
Anomalie / Bug	Non-conformité entre résultat obtenu et résultat attendu
Test fonctionnel	Valider que la fonction se comporte correctement
Test d'intégration	Vérifier communication entre équipements
Test de performance	Vérifier temps de réponse, capacité du système
Test de sécurité	Contrôle des accès, cryptage, vulnérabilités
Test de conformité	Respect d'une norme ou cahier des charges
Test utilisateur (UAT)	Validation finale par le client

Partie VII – Méthodologie de rédaction d'un plan de tests

1. Étapes de création

Phase 1 : Analyse (1-2 jours)

- Lecture et compréhension du cahier des charges
- Identification des fonctionnalités à tester

- Définition du périmètre de test
- Identification des risques

Phase 2 : Conception (3-5 jours)

- Création de la structure du plan de tests
- Rédaction des cas de tests détaillés
- Définition des critères de réussite
- Organisation des tests par priorité

Phase 3 : Révision (1-2 jours)

- Validation par l'équipe projet
- Ajustements selon retours
- Finalisation du document

Phase 4 : Exécution (selon planning)

- Exécution des tests selon le plan
- Enregistrement des résultats
- Suivi des anomalies

2. Conseils pratiques

Pour rédiger un bon cas de test :

- □ Être précis et détaillé dans les procédures
- □ Définir des critères de réussite mesurables
- □ Prévoir les préconditions nécessaires
- □ Anticiper les cas limites
- □ Rédiger de manière claire et compréhensible

Erreurs à éviter :

- □ Procédures trop vagues ("Vérifier que ça marche")
- □ Critères de réussite subjectifs ("Interface agréable")
- □ Oubli des préconditions
- □ Tests non reproductibles
- □ Manque de traçabilité avec les exigences

3. Outils de gestion

Tableurs (Excel, Google Sheets) :

- Avantages : Facile à utiliser, accessible à tous
- Inconvénients : Pas de traçabilité automatique, gestion de versions difficile

Outils dédiés (TestLink, Squash, Jira) :

- Avantages : Traçabilité, gestion des versions, reporting automatique
- Inconvénients : Courbe d'apprentissage, coût éventuel

Recommandation : Pour débuter, un tableur est suffisant. Pour des projets complexes, un

outil dédié est préférable.

Partie VIII – Éléments de synthèse

Un bon plan de test :

- **Couverture complète** : Couvre 100% des fonctionnalités critiques
- **Méthodologie structurée** : Suit une méthodologie documentée et reproductible
- **Protection contre les régressions** : Permet de détecter les régressions après modifications
- **Confiance client** : Améliore la confiance du client dans le système
- **Traçabilité totale** : Chaque test peut être relié à une exigence
- **Documentation complète** : Procédures claires et résultats documentés
- **Adaptabilité** : Peut être mis à jour selon l'évolution du projet

Sans plan de tests :

- **Risques opérationnels** : Risque de panne en exploitation
- **Coûts supplémentaires** : Surcoûts de maintenance et corrections tardives
- **Image dégradée** : Perte de confiance du client
- **Reprise tardive** : Erreurs découvertes trop tard, corrections coûteuses
- **Manque de traçabilité** : Difficulté à prouver la conformité
- **Formation difficile** : Pas de documentation pour former les nouveaux utilisateurs

Bénéfices mesurables

Retour sur investissement :

- Réduction de 60-80% des bugs en production
- Diminution de 40-50% des coûts de maintenance
- Amélioration de 30-50% de la satisfaction client
- Réduction de 50-70% du temps de mise en service

Indicateurs de qualité :

- Taux de réussite des tests > 95%
- Couverture fonctionnelle > 90%
- Temps moyen de résolution des anomalies < 3 jours
- Nombre de régressions < 5% après corrections

Conclusion

Le test est une phase stratégique, pas une formalité.

Un système BMS/GTB bien testé garantit :

- **Fiabilité opérationnelle** : Le système fonctionne comme prévu en conditions

réelles
- **Performance énergétique** : Optimisation de la consommation énergétique
- **Satisfaction utilisateur** : Interface intuitive et fonctionnalités adaptées
- **Conformité aux normes** : Respect des réglementations et standards
- **Réduction des coûts** : Moins de pannes, moins de maintenance corrective
- **Sécurité** : Protection contre les accès non autorisés et les pannes critiques

La gestion des plans de tests est donc un élément essentiel dans la réussite d'un projet de bâtiment intelligent.

Investir du temps dans la conception et l'exécution d'un plan de tests complet permet d'économiser du temps et de l'argent sur le long terme, tout en garantissant la qualité et la fiabilité du système livré.

Ressources à Télécharger

Étude de Cas - Plan de Tests BMS

Télécharger l'étude de cas complète (PDF)

Ce document contient un cahier des charges détaillé pour créer un plan de tests complet sur un système de gestion technique de bâtiment.

Modèle de Plan de Tests

Télécharger le modèle de plan de tests (DOCX)

Ce modèle simplifié vous permet de créer votre propre plan de tests avec des templates prêts à remplir.