

# Ingénierie système, MBSE & V&V

Plus un système devient complexe, moins il est possible de l'optimiser en améliorant chaque **sous-système** séparément. Une batterie plus grande change masse et thermique; une antenne plus puissante change énergie et pointage; une nouvelle procédure change logiciel, formation et opérations. L'ingénierie système organise ces dépendances depuis les besoins des parties prenantes jusqu'à la preuve que le produit réalisé répond réellement à la mission.

## Comprendre le problème avant les équations

Question de départ — Ingénierie système, MBSE & V&V: comment « Comprendre avant de calculer » agit-il sur « exigence », et quel repère issu de « traçabilité » permet de vérifier la décision?

**Intuition.** Une équipe doit prouver qu'un habitat répond au besoin « maintenir quatre personnes en sécurité pendant une panne de ravitaillement », sans laisser ce besoin se perdre entre dizaines de sous-systèmes.

1

### Comprendre avant de calculer

Une équipe doit prouver qu'un habitat répond au besoin « maintenir quatre personnes en sécurité pendant une panne de ravitaillement », sans laisser ce besoin se perdre entre dizaines de sous-systèmes.

2

### exigence

Définition sans prérequis.

3

### traçabilité

Définition sans prérequis.

**Repère concret.** « Le système doit être fiable » est trop vague; « le système doit fournir au moins X litres d'eau potable par personne et par jour dans tel mode » devient testable.

- Comprendre avant de calculer — Une équipe doit prouver qu'un habitat répond au besoin « maintenir quatre personnes en sécurité pendant une panne de ravitaillement », sans laisser ce besoin se perdre entre...
- exigence — Définition sans prérequis.
- traçabilité — Définition sans prérequis.

Ce module s'appuie sur modules 01 à 09 recommandés, mais il reste lisible isolément: les symboles déterminants pour ingénierie système, mbse & v&v sont explicités au premier calcul.

RAMPE ZÉRO PRÉREQUIS · MODULE 21

## Comprendre avant de calculer

Une équipe doit prouver qu'un habitat répond au besoin « maintenir quatre personnes en sécurité pendant une panne de ravitaillement », sans laisser ce besoin se perdre entre dizaines de sous-systèmes.

Progression du module — exigence; interface; vérification; traçabilité.

### exigence

**Définition sans prérequis.** Une **exigence** décrit une propriété ou une performance vérifiable que le système doit satisfaire. Une bonne exigence précise quoi, dans quelles conditions et selon quel critère.

**Scène concrète.** « Le système doit être fiable » est trop vague; « le système doit fournir au moins X litres d'eau potable par personne et par jour dans tel mode » devient testable.

**Contrôle mental.** Une **exigence** vérifiable doit contenir une condition et un critère observable; si deux lecteurs peuvent accepter des résultats différents, elle n'est pas assez précise.

**Piège à éviter.** Une phrase ambitieuse n'est pas une exigence si personne ne peut démontrer objectivement qu'elle est satisfaite.

**Pourquoi cela compte en mission.** Dans ce module, une erreur sur « [exigence](#) » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

## Exercice guidé — exigence

**Situation à reconnaître.** « Le système doit être fiable » est trop vague; « le système doit fournir au moins X litres d'eau potable par personne et par jour dans tel mode » devient testable.

**Vérification demandée.** Une exigence vérifiable doit contenir une condition et un critère observable; si deux lecteurs peuvent accepter des résultats différents, elle n'est pas assez précise.

**Erreur à écarter.** Une phrase ambitieuse n'est pas une exigence si personne ne peut démontrer objectivement qu'elle est satisfaite.

## Corrigé raisonné

Sens précis

Une exigence décrit une propriété ou une performance vérifiable que le système doit satisfaire. Une bonne exigence précise quoi, dans quelles conditions et selon quel critère.

Test du cas

Une exigence vérifiable doit contenir une condition et un critère observable; si deux lecteurs peuvent accepter des résultats différents, elle n'est pas assez précise.

Piège exclu

Une phrase ambitieuse n'est pas une exigence si personne ne peut démontrer objectivement qu'elle est satisfaite.

Conséquence opérationnelle

Dans ce module, une erreur sur « exigence » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

## interface

**Définition sans prérequis.** Une [interface](#) est une frontière où deux éléments échangent matière, énergie, données, forces, chaleur ou responsabilité opérationnelle.

**Scène concrète.** Un sas possède des interfaces mécaniques, électriques, fluides, logicielles et humaines avec l'habitat et le scaphandre.

**Contrôle mental.** Pour chaque [interface](#), nommez ce qui traverse la frontière, dans quel sens, avec quelle unité ou protocole et qui en est propriétaire.

**Piège à éviter.** Les échecs d'intégration apparaissent souvent aux interfaces, même lorsque chaque composant fonctionne correctement seul.

**Pourquoi cela compte en mission.** Dans ce module, une erreur sur « [interface](#) » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

## Exercice guidé — interface

**Situation à reconnaître.** Un sas possède des interfaces mécaniques, électriques, fluides, logicielles et humaines avec l'habitat et le scaphandre.

**Vérification demandée.** Pour chaque interface, nommez ce qui traverse la frontière, dans quel sens, avec quelle unité ou protocole et qui en est propriétaire.

**Erreur à écarter.** Les échecs d'intégration apparaissent souvent aux interfaces, même lorsque chaque composant fonctionne correctement seul.

## Corrigé raisonné

Sens précis

Une interface est une frontière où deux éléments échangent matière, énergie, données, forces, chaleur ou responsabilité opérationnelle.

Test du cas

Pour chaque interface, nommez ce qui traverse la frontière, dans quel sens, avec quelle unité ou protocole et qui en est propriétaire.

Piège exclu

Les échecs d'intégration apparaissent souvent aux interfaces, même lorsque chaque composant fonctionne correctement seul.

Conséquence opérationnelle

Dans ce module, une erreur sur « interface » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

**Termes du corrigé.** [observation](#)

Quantification

interface: relation qualitative ici; aucune unité intrinsèque.

Vérification

interface: confronter conclusion, contrôle et piège de la carte.

## vérification

**Définition sans prérequis.** La **vérification** répond à la question « avons-nous construit le système conformément aux exigences? » à l'aide d'essais, analyses, inspections ou démonstrations.

**Scène concrète.** Une exigence de tenue à la pression peut être vérifiée par analyse structurale et par essai de qualification selon le plan choisi.

**Contrôle mental.** Demandez quelle preuve montrerait que l'exigence est satisfaite: test, analyse, inspection ou démonstration. Sans preuve définie, la **vérification** reste une intention.

**Piège à éviter.** Vérifier une pièce n'est pas valider la mission: la validation traite l'adéquation du système réel au besoin d'usage.

**Pourquoi cela compte en mission.** Dans ce module, une erreur sur « **vérification** » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

## Exercice guidé — vérification

**Situation à reconnaître.** Une exigence de tenue à la pression peut être vérifiée par analyse structurale et par essai de qualification selon le plan choisi.

**Vérification demandée.** Demandez quelle preuve montrerait que l'exigence est satisfaite: test, analyse, inspection ou démonstration. Sans preuve définie, la vérification reste une intention.

**Erreur à écarter.** Vérifier une pièce n'est pas valider la mission: la validation traite l'adéquation du système réel au besoin d'usage.

## Corrigé raisonné

Sens précis

La vérification répond à la question « avons-nous construit le système conformément aux exigences? » à l'aide d'essais, analyses, inspections ou démonstrations.

Test du cas

Demandez quelle preuve montrerait que l'exigence est satisfaite: test, analyse, inspection ou démonstration. Sans preuve définie, la vérification reste une intention.

Piège exclu

Vérifier une pièce n'est pas valider la mission: la validation traite l'adéquation du système réel au besoin d'usage.

Conséquence opérationnelle

Dans ce module, une erreur sur « **vérification** » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

Quantification

vérification: relation qualitative ici; aucune unité intrinsèque.

Vérification

vérification: confronter conclusion, contrôle et piège de la carte.

## traçabilité

**Définition sans prérequis.** La **traçabilité** relie besoins, exigences, architecture, analyses, essais, anomalies et preuves. Elle permet de retrouver pourquoi une décision existe et ce qui doit être retesté après modification.

**Scène concrète.** Si un capteur change, la traçabilité aide à identifier les exigences, logiciels et essais affectés.

**Contrôle mental.** Choisissez une exigence et remontez jusqu'au besoin puis descendez jusqu'au test: toute rupture dans cette chaîne indique une décision qui ne peut plus être justifiée.

**Piège à éviter.** Une matrice de traçabilité remplie après coup sans liens vivants avec les décisions n'apporte qu'une illusion de contrôle.

**Pourquoi cela compte en mission.** Dans ce module, une erreur sur « **traçabilité** » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

## Exercice guidé — traçabilité

**Situation à reconnaître.** Si un capteur change, la traçabilité aide à identifier les exigences, logiciels et essais affectés.

**Vérification demandée.** Choisissez une exigence et remontez jusqu'au besoin puis descendez jusqu'au test: toute rupture dans cette chaîne indique une décision qui ne peut plus être justifiée.

**Erreur à écarter.** Une matrice de traçabilité remplie après coup sans liens vivants avec les décisions n'apporte qu'une illusion de contrôle.

## Corrigé raisonné

### Sens précis

La traçabilité relie besoins, exigences, architecture, analyses, essais, anomalies et preuves. Elle permet de retrouver pourquoi une décision existe et ce qui doit être retesté après modification.

### Test du cas

Choisissez une exigence et remontez jusqu'au besoin puis descendez jusqu'au test: toute rupture dans cette chaîne indique une décision qui ne peut plus être justifiée.

### Piège exclu

Une matrice de traçabilité remplie après coup sans liens vivants avec les décisions n'apporte qu'une illusion de contrôle.

### Conséquence opérationnelle

Dans ce module, une erreur sur « traçabilité » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé.

### Quantification

traçabilité: relation qualitative ici; aucune unité intrinsèque.

### Vérification

traçabilité: confronter conclusion, contrôle et piège de la carte.

## Étude intégrée — Ingénierie système, MBSE & V&V

**Situation à analyser.** Une équipe doit prouver qu'un habitat répond au besoin « maintenir quatre personnes en sécurité pendant une panne de ravitaillement », sans laisser ce besoin se perdre entre dizaines de sous-systèmes.

- **exigence:** « Le système doit être fiable » est trop vague; « le système doit fournir au moins X litres d'eau potable par personne et par jour dans tel mode » devient testable. Contrôle attendu: Une exigence vérifiable doit contenir une condition et un critère observable; si deux lecteurs peuvent accepter des résultats différents, elle n'est pas assez précise.
- **interface:** Un sas possède des interfaces mécaniques, électriques, fluides, logicielles et humaines avec l'habitat et le scaphandre. Contrôle attendu: Pour chaque interface, nommez ce qui traverse la frontière, dans quel sens, avec quelle unité ou protocole et qui en est propriétaire.
- **vérification:** Une exigence de tenue à la pression peut être vérifiée par analyse structurale et par essai de qualification selon le plan choisi. Contrôle attendu: Demandez quelle preuve montrerait que l'exigence est satisfaite: test, analyse, inspection ou démonstration. Sans preuve définie, la vérification reste une intention.
- **traçabilité:** Si un capteur change, la traçabilité aide à identifier les exigences, logiciels et essais affectés. Contrôle attendu: Choisissez une exigence et remontez jusqu'au besoin puis descendez jusqu'au test: toute rupture dans cette chaîne indique une décision qui ne peut plus être justifiée.

Le diagnostic ne doit pas confondre « exigence » et « interface ». Le premier se reconnaît ici par le cas suivant: « Le système doit être fiable » est trop vague; « le système doit fournir au moins X litres d'eau potable par personne et par jour dans tel mode » devient testable. Le second doit être contrôlé autrement: Pour chaque interface, nommez ce qui traverse la frontière, dans quel sens, avec quelle unité ou protocole et qui en est propriétaire.

Une seconde vérification croise « vérification » et « traçabilité ». Gardez comme alerte « Vérifier une pièce n'est pas valider la mission: la validation traite l'adéquation du système réel au besoin d'usage. »; pour « traçabilité », utilisez plutôt ce test: Choisissez une exigence et remontez jusqu'au besoin puis descendez jusqu'au test: toute rupture dans cette chaîne indique une décision qui ne peut plus être justifiée.

## Exercice de synthèse

Dans la situation de « Ingénierie système, MBSE & V&V », classez les indices selon « exigence » (Une exigence vérifiable doit contenir une condition et un critère observable; si deux lecteurs peuvent accepter des résultats différents, elle n'est pas assez précise.), « interface » (Pour chaque interface, nommez ce qui traverse la frontière, dans quel sens, avec quelle unité ou protocole et qui en est propriétaire.), « vérification » (Demandez quelle preuve montrerait que l'exigence est satisfaite: test, analyse, inspection ou démonstration. Sans preuve définie, la vérification reste une intention.), « traçabilité » (Choisissez une exigence et remontez jusqu'au besoin puis descendez jusqu'au test: toute rupture dans cette chaîne indique une décision qui ne peut plus être justifiée.) Décision suspendue si un contrôle échoue.

## Corrigé de synthèse

Pour exigence, Dans ce module, une erreur sur « exigence » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé. Le garde-fou principal est: Une phrase ambitieuse n'est pas une exigence si personne ne peut démontrer objectivement qu'elle est satisfaite. Pour interface, Dans ce module, une erreur sur « interface » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé. Le garde-fou principal est: Les échecs d'intégration apparaissent souvent aux interfaces, même lorsque chaque composant fonctionne correctement seul. Pour vérification, Dans ce module, une erreur sur « vérification » peut laisser une exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé. Le garde-fou principal est: Vérifier une pièce n'est pas valider la mission: la validation traite l'adéquation du système réel au besoin d'usage. Pour traçabilité, Dans ce module, une erreur sur « traçabilité » peut laisser une

exigence sans preuve, une interface sans propriétaire ou une modification sans impact analysé. Le garde-fou principal est: Une matrice de traçabilité remplie après coup sans liens vivants avec les décisions n'apporte qu'une illusion de contrôle.

## Mini-leçons quantitatives canoniques

Ce bloc constitue la référence quantitative du module; les valeurs d'exercice sont pédagogiques et ne doivent pas être confondues avec des données de mission.

### Couverture de vérification

$$C\_verify = N\_accepted / N\_applicable$$

1 — Question concrète

Que permet de calculer «  $C\_verify = N\_accepted / N\_applicable$  » dans « Couverture de vérification » ?

2 — Intuition sans symboles

La couverture indique quelle part des obligations de vérification possède déjà une preuve acceptée.

3 — Grandeurs

$C\_verify$ : couverture de vérification;  $N\_accepted$ : exigences fermées par preuve acceptée;  $N\_applicable$ : exigences applicables

4 — Formule

$$C\_verify = N\_accepted / N\_applicable$$

5 — Lecture

Lire «  $C\_verify = N\_accepted / N\_applicable$  » en nommant chaque opération et chaque indice.

6 — Symboles et sens

$C\_verify$ : couverture de vérification;  $N\_accepted$ : exigences fermées par preuve acceptée;  $N\_applicable$ : exigences applicables

7 — Prononciation

Dire explicitement les indices, puissances, lettres grecques et parenthèses afin de reconstruire la relation sans voir la page.

8 — Unités

comptes sans dimension;  $C\_verify$  exprimable en %

9 — Convention

Conserver le même repère, la même base de temps, la même frontière de système et la même convention de signe pendant toute la substitution.

10 — Pourquoi cette opération

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

11 — Hypothèses

Les deux quantités doivent appartenir au même périmètre et à la même période.

12 — Contrôle d'unités

comptes sans dimension;  $C\_verify$  exprimable en % Vérifier que la simplification dimensionnelle conduit bien à l'unité de la grandeur recherchée.

13 — Cas numérique

Avec  $N\_accepted=171$  et  $N\_applicable=180$ ,  $C\_verify=0,95=95\%$ .

14 — Pourquoi le calcul fonctionne

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

15 — Contrôle indépendant

Multiplier le résultat par le dénominateur doit restituer le numérateur.

16 — Estimation mentale

Avant calcul, comparer mentalement le numérateur au dénominateur pour savoir si le résultat doit être inférieur, proche ou supérieur à un.

17 — Interprétation

Une couverture élevée ne dit rien sur la qualité des preuves restantes ni sur leur criticité.

18 — Ce que le résultat ne prouve pas

Le calcul ne prouve ni que les entrées sont justes ni que le modèle reste valide hors du scénario déclaré; ces deux questions doivent être vérifiées séparément.

19 — Sensibilité

Le résultat augmente si le numérateur augmente et diminue si le dénominateur augmente, toutes choses égales par ailleurs.

20 — Exercices guidé et autonome

**Exercice guidé.**  $N\_accepted=284$ ,  $N\_applicable=320$ .

**Correction guidée détaillée — ouvrir après essai**

N\_accepted=284, N\_applicable=320. C\_verify=0,8875=88,75 %.

**Exercice autonome.** N\_accepted=95, N\_applicable=100.

**Correction autonome — ouvrir après essai**

N\_accepted=95, N\_applicable=100. C\_verify=95 %.

21 — Décision mission

Fermer en priorité les exigences critiques et les preuves bloquantes, pas seulement maximiser le pourcentage.

## Marge absolue de masse

$M_{\text{mass}} = M_{\text{allocated}} - M_{\text{estimated}}$

1 — Question concrète

Que permet de calculer «  $M_{\text{mass}} = M_{\text{allocated}} - M_{\text{estimated}}$  » dans « Marge absolue de masse » ?

2 — Intuition sans symboles

La marge de masse mesure l'espace restant dans une allocation après l'estimation actuelle.

3 — Grandeurs

$M_{\text{mass}}$ : marge de masse;  $M_{\text{allocated}}$ : allocation maximale;  $M_{\text{estimated}}$ : masse estimée actuelle

4 — Formule

$M_{\text{mass}} = M_{\text{allocated}} - M_{\text{estimated}}$

5 — Lecture

Lire «  $M_{\text{mass}} = M_{\text{allocated}} - M_{\text{estimated}}$  » en nommant chaque opération et chaque indice.

6 — Symboles et sens

$M_{\text{mass}}$ : marge de masse;  $M_{\text{allocated}}$ : allocation maximale;  $M_{\text{estimated}}$ : masse estimée actuelle

7 — Prononciation

Dire explicitement les indices, puissances, lettres grecques et parenthèses afin de reconstruire la relation sans voir la page.

8 — Unités

toutes les masses dans la même unité

9 — Convention

Conserver le même repère, la même base de temps, la même frontière de système et la même convention de signe pendant toute la substitution.

10 — Pourquoi cette opération

La soustraction isole la marge ou la variation entre deux grandeurs exprimées sur la même base.

11 — Hypothèses

Les deux termes doivent utiliser le même repère, la même unité et la même frontière de système.

12 — Contrôle d'unités

toutes les masses dans la même unité Vérifier que la simplification dimensionnelle conduit bien à l'unité de la grandeur recherchée.

13 — Cas numérique

Avec  $M_{\text{allocated}}=1\,000\text{ kg}$  et  $M_{\text{estimated}}=860\text{ kg}$ ,  $M_{\text{mass}}=140\text{ kg}$ .

14 — Pourquoi le calcul fonctionne

La soustraction isole la marge ou la variation entre deux grandeurs exprimées sur la même base.

15 — Contrôle indépendant

Ajouter la différence au terme retiré doit restituer le terme initial.

16 — Estimation mentale

Comparer les deux termes suffit à prévoir le signe et l'ordre de grandeur de la différence.

17 — Interprétation

Une marge positive peut encore être insuffisante si l'estimation est immature ou si des réserves doivent être protégées.

18 — Ce que le résultat ne prouve pas

Le calcul ne prouve ni que les entrées sont justes ni que le modèle reste valide hors du scénario déclaré; ces deux questions doivent être vérifiées séparément.

19 — Sensibilité

Une hausse du premier terme augmente le résultat; une hausse du second le diminue à pente unitaire.

20 — Exercices guidé et autonome

**Exercice guidé.**  $M_{\text{allocated}}=500\text{ kg}$ ,  $M_{\text{estimated}}=455\text{ kg}$ .

**Correction guidée détaillée — ouvrir après essai**

$M_{\text{allocated}}=500\text{ kg}$ ,  $M_{\text{estimated}}=455\text{ kg}$ .  $M_{\text{mass}}=45\text{ kg}$ .

**Exercice autonome.**  $M_{\text{allocated}}=2\,000\text{ kg}$ ,  $M_{\text{estimated}}=1\,920\text{ kg}$ .

**Correction autonome — ouvrir après essai**

$M_{\text{allocated}}=2\,000\text{ kg}$ ,  $M_{\text{estimated}}=1\,920\text{ kg}$ .  $M_{\text{mass}}=80\text{ kg}$ .

21 — Décision mission

Associer toute marge à la maturité, aux risques et à la politique de réserve du programme.

## Marge relative à l'allocation

$$M_{\text{pct}} = 100 \times M_{\text{mass}} / M_{\text{allocated}}$$

1 — Question concrète

Que permet de calculer «  $M_{\text{pct}} = 100 \times M_{\text{mass}} / M_{\text{allocated}}$  » dans « Marge relative à l'allocation » ?

2 — Intuition sans symboles

Exprimer la marge relativement à l'allocation permet de comparer des sous-systèmes de tailles différentes.

3 — Grandeurs

$M_{\text{pct}}$ : marge en pourcentage;  $M_{\text{mass}}$ : marge absolue;  $M_{\text{allocated}}$ : allocation maximale

4 — Formule

$$M_{\text{pct}} = 100 \times M_{\text{mass}} / M_{\text{allocated}}$$

5 — Lecture

Lire «  $M_{\text{pct}} = 100 \times M_{\text{mass}} / M_{\text{allocated}}$  » en nommant chaque opération et chaque indice.

6 — Symboles et sens

$M_{\text{pct}}$ : marge en pourcentage;  $M_{\text{mass}}$ : marge absolue;  $M_{\text{allocated}}$ : allocation maximale

7 — Prononciation

Dire explicitement les indices, puissances, lettres grecques et parenthèses afin de reconstruire la relation sans voir la page.

8 — Unités

$M_{\text{pct}}$  en %; masses dans la même unité

9 — Convention

Conserver le même repère, la même base de temps, la même frontière de système et la même convention de signe pendant toute la substitution.

10 — Pourquoi cette opération

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

11 — Hypothèses

Les deux quantités doivent appartenir au même périmètre et à la même période.

12 — Contrôle d'unités

$M_{\text{pct}}$  en %; masses dans la même unité Vérifier que la simplification dimensionnelle conduit bien à l'unité de la grandeur recherchée.

13 — Cas numérique

Avec  $M_{\text{mass}}=140\text{ kg}$  et  $M_{\text{allocated}}=1\,000\text{ kg}$ ,  $M_{\text{pct}}=14\%$ .

14 — Pourquoi le calcul fonctionne

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

15 — Contrôle indépendant

Multiplier le résultat par le dénominateur doit restituer le numérateur.

16 — Estimation mentale

Avant calcul, comparer mentalement le numérateur au dénominateur pour savoir si le résultat doit être inférieur, proche ou supérieur à un.

17 — Interprétation

Le pourcentage dépend du dénominateur choisi; il faut toujours nommer explicitement la base.

18 — Ce que le résultat ne prouve pas

Le calcul ne prouve ni que les entrées sont justes ni que le modèle reste valide hors du scénario déclaré; ces deux questions doivent être vérifiées séparément.

19 — Sensibilité

Le résultat augmente si le numérateur augmente et diminue si le dénominateur augmente, toutes choses égales par ailleurs.

20 — Exercices guidé et autonome

**Exercice guidé.**  $M_{\text{mass}}=45\text{ kg}$ ,  $M_{\text{allocated}}=500\text{ kg}$ .

**Correction guidée détaillée — ouvrir après essai**

$M_{\text{mass}}=45\text{ kg}$ ,  $M_{\text{allocated}}=500\text{ kg}$ .  $M_{\text{pct}}=9\%$ .

**Exercice autonome.**  $M_{\text{mass}}=80$  kg,  $M_{\text{allocated}}=2\,000$  kg.

**Correction autonome — ouvrir après essai**

$M_{\text{mass}}=80$  kg,  $M_{\text{allocated}}=2\,000$  kg.  $M_{\text{pct}}=4$  %.

21 — Décision mission

Comparer les marges seulement si leur définition et leur base sont identiques.

## Couverture de traçabilité

$C_{\text{trace}} = N_{\text{traced}} / N_{\text{total}}$

1 — Question concrète

Que permet de calculer «  $C_{\text{trace}} = N_{\text{traced}} / N_{\text{total}}$  » dans « Couverture de traçabilité » ?

2 — Intuition sans symboles

La traçabilité n'est complète que si chaque exigence peut remonter au besoin et descendre vers une preuve.

3 — Grandeurs

$C_{\text{trace}}$ : couverture de traçabilité;  $N_{\text{traced}}$ : exigences reliées à une source et une preuve;  $N_{\text{total}}$ : exigences du périmètre

4 — Formule

$C_{\text{trace}} = N_{\text{traced}} / N_{\text{total}}$

5 — Lecture

Lire «  $C_{\text{trace}} = N_{\text{traced}} / N_{\text{total}}$  » en nommant chaque opération et chaque indice.

6 — Symboles et sens

$C_{\text{trace}}$ : couverture de traçabilité;  $N_{\text{traced}}$ : exigences reliées à une source et une preuve;  $N_{\text{total}}$ : exigences du périmètre

7 — Prononciation

Dire explicitement les indices, puissances, lettres grecques et parenthèses afin de reconstruire la relation sans voir la page.

8 — Unités

comptes sans dimension;  $C_{\text{trace}}$  exprimable en %

9 — Convention

Conserver le même repère, la même base de temps, la même frontière de système et la même convention de signe pendant toute la substitution.

10 — Pourquoi cette opération

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

11 — Hypothèses

Les deux quantités doivent appartenir au même périmètre et à la même période.

12 — Contrôle d'unités

comptes sans dimension;  $C_{\text{trace}}$  exprimable en % Vérifier que la simplification dimensionnelle conduit bien à l'unité de la grandeur recherchée.

13 — Cas numérique

Avec  $N_{\text{traced}}=240$  et  $N_{\text{total}}=250$ ,  $C_{\text{trace}}=0,96=96$  %.

14 — Pourquoi le calcul fonctionne

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

15 — Contrôle indépendant

Multiplier le résultat par le dénominateur doit restituer le numérateur.

16 — Estimation mentale

Avant calcul, comparer mentalement le numérateur au dénominateur pour savoir si le résultat doit être inférieur, proche ou supérieur à un.

17 — Interprétation

Un lien présent mais faux ou obsolète n'est pas une traçabilité de qualité.

18 — Ce que le résultat ne prouve pas

Le calcul ne prouve ni que les entrées sont justes ni que le modèle reste valide hors du scénario déclaré; ces deux questions doivent être vérifiées séparément.

19 — Sensibilité

Le résultat augmente si le numérateur augmente et diminue si le dénominateur augmente, toutes choses égales par ailleurs.

20 — Exercices guidé et autonome

**Exercice guidé.**  $N_{\text{traced}}=178$ ,  $N_{\text{total}}=200$ .

**Correction guidée détaillée — ouvrir après essai**

N\_traced=178, N\_total=200. C\_trace=89 %.

**Exercice autonome.** N\_traced=99, N\_total=100.

**Correction autonome — ouvrir après essai**

N\_traced=99, N\_total=100. C\_trace=99 %.

21 — Décision mission

Auditer sémantiquement les liens critiques avant une revue de configuration.

## Fermeture des interfaces

$C_{\text{interface}} = N_{\text{closed}} / N_{\text{total}}$

1 — Question concrète

Que permet de calculer «  $C_{\text{interface}} = N_{\text{closed}} / N_{\text{total}}$  » dans « Fermeture des interfaces » ?

2 — Intuition sans symboles

Les interfaces ouvertes concentrent souvent des risques qui n'apparaissent pas dans les performances propres de chaque sous-système.

3 — Grandeurs

$C_{\text{interface}}$ : taux de fermeture;  $N_{\text{closed}}$ : interfaces techniquement fermées;  $N_{\text{total}}$ : interfaces du périmètre

4 — Formule

$C_{\text{interface}} = N_{\text{closed}} / N_{\text{total}}$

5 — Lecture

Lire «  $C_{\text{interface}} = N_{\text{closed}} / N_{\text{total}}$  » en nommant chaque opération et chaque indice.

6 — Symboles et sens

$C_{\text{interface}}$ : taux de fermeture;  $N_{\text{closed}}$ : interfaces techniquement fermées;  $N_{\text{total}}$ : interfaces du périmètre

7 — Prononciation

Dire explicitement les indices, puissances, lettres grecques et parenthèses afin de reconstruire la relation sans voir la page.

8 — Unités

comptes sans dimension;  $C_{\text{interface}}$  exprimable en %

9 — Convention

Conserver le même repère, la même base de temps, la même frontière de système et la même convention de signe pendant toute la substitution.

10 — Pourquoi cette opération

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

11 — Hypothèses

Les deux quantités doivent appartenir au même périmètre et à la même période.

12 — Contrôle d'unités

comptes sans dimension;  $C_{\text{interface}}$  exprimable en % Vérifier que la simplification dimensionnelle conduit bien à l'unité de la grandeur recherchée.

13 — Cas numérique

Avec  $N_{\text{closed}}=45$  et  $N_{\text{total}}=50$ ,  $C_{\text{interface}}=0,90=90\%$ .

14 — Pourquoi le calcul fonctionne

La division compare une quantité à une référence homogène et produit un indicateur de proportion.

15 — Contrôle indépendant

Multiplier le résultat par le dénominateur doit restituer le numérateur.

16 — Estimation mentale

Avant calcul, comparer mentalement le numérateur au dénominateur pour savoir si le résultat doit être inférieur, proche ou supérieur à un.

17 — Interprétation

Le taux ne distingue pas une interface mineure d'une interface vitale; le registre de criticité reste indispensable.

18 — Ce que le résultat ne prouve pas

Le calcul ne prouve ni que les entrées sont justes ni que le modèle reste valide hors du scénario déclaré; ces deux questions doivent être vérifiées séparément.

19 — Sensibilité

Le résultat augmente si le numérateur augmente et diminue si le dénominateur augmente, toutes choses égales par ailleurs.

20 — Exercices guidé et autonome

**Exercice guidé.**  $N_{\text{closed}}=72$ ,  $N_{\text{total}}=80$ .

### Correction guidée détaillée — ouvrir après essai

N\_closed=72, N\_total=80. C\_interface=90 %.

**Exercice autonome.** N\_closed=38, N\_total=40.

### Correction autonome — ouvrir après essai

N\_closed=38, N\_total=40. C\_interface=95 %.

21 — Décision mission

Refuser de déclarer l'architecture mature tant qu'une interface critique reste non définie ou non vérifiée.

## Progression de maîtrise

1. exigence: appliquez le contrôle concret de sa fiche.
2. interface: appliquez le contrôle concret de sa fiche.
3. vérification: appliquez le contrôle concret de sa fiche.
4. traçabilité: appliquez le contrôle concret de sa fiche.
5. Synthèse de « Ingénierie système, MBSE & V&V »: confrontez « exigence » à « interface »; choisissez le contrôle prioritaire.

## Vérification de maîtrise

Validation 21 — exigence, interface, vérification, traçabilité. Pour chaque notion: un indice et un contrôle.

## Objectifs de maîtrise

- Savoir reconnaître un cas réel de « Comprendre avant de calculer » à partir d'indices observables.
- Présenter les données et unités nécessaires à l'analyse de « exigence ».
- Tester la sensibilité du résultat à une hypothèse de « interface ».
- Formuler le critère qui autorise, retarde ou interdit la décision liée à « traçabilité ».

## 1. Besoin, objectif, exigence: trois niveaux à ne pas confondre

Un besoin exprime pourquoi une capacité est nécessaire. Un objectif décrit ce que la mission cherche à accomplir. Une exigence impose une propriété vérifiable au système. « Survivre sur Mars » n'est pas une bonne exigence; « maintenir la pression de l'habitat dans une plage définie pendant telle durée après une panne simple » peut le devenir. Une exigence claire possède un sujet, une action, une condition et un critère mesurable.

**Réflexe d'ingénierie.** Pour « besoin, objectif, exigence: trois niveaux à ne pas confondre », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « module 21 · Réflexe d'ingénierie ».

### Approfondissement — Une exigence n'est vérifiable que si son verbe produit une preuve

« Le système devra être robuste » n'est pas une exigence utilisable parce qu'aucun test ne peut répondre sans interprétation. Une bonne exigence précise une fonction, une condition et un critère. Par exemple: « Après la perte d'un convertisseur, le sous-système ECLSS doit maintenir une **puissance** disponible d'au moins 8 kW pendant 30 min. » Ici, 8 kW est la puissance minimale et 30 min la durée. La vérification peut alors être planifiée par essai, analyse, inspection ou démonstration.

La matrice de vérification relie chaque exigence à sa méthode, son responsable, son niveau d'essai et son résultat. Sans cette traçabilité, une équipe peut accumuler des centaines de tests sans savoir si toutes les exigences ont réellement été couvertes. Sur un habitat martien, c'est particulièrement dangereux: les sous-systèmes évoluent pendant des années et une exigence de survie peut être perdue lors d'un changement de fournisseur ou d'architecture.

## 2. ConOps: raconter comment le système sera réellement utilisé

Le Concept of Operations décrit acteurs, phases, environnements, flux d'information, états et scénarios nominaux ou dégradés. Il évite de concevoir une machine sans comprendre le travail humain autour d'elle. Une excellente exigence technique peut être inutile si le scénario d'exploitation est faux. Le **ConOps** est donc un pont entre mission, opérations et architecture.

**Réflexe d'ingénierie.** Pour « conops: raconter comment le système sera réellement utilisé », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Une exigence n'est

vérifiable ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Une exigence n'est vérifiable ».

### 3. Décomposition fonctionnelle et architecture

---

On part des fonctions nécessaires puis on décide quels éléments physiques ou logiciels les réalisent. Cette séparation empêche de choisir trop tôt une solution favorite. Une fonction « rejeter la chaleur » peut être réalisée par différentes combinaisons de radiateurs, boucles et modes d'exploitation. L'architecture alloue ensuite fonctions, performances et interfaces aux sous-systèmes.

**Réflexe d'ingénierie.** Pour « décomposition fonctionnelle et architecture », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Une exigence n'est vérifiable ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Une exigence n'est vérifiable ».

### 4. Interfaces: là où les bons sous-systèmes peuvent devenir un mauvais système

---

Une interface ne se limite pas à un connecteur. Elle inclut mécanique, énergie, données, thermique, logiciel, procédures et responsabilités. Les unités, tolérances, états autorisés et comportement en panne doivent être définis. Les problèmes d'intégration viennent souvent de suppositions implicites faites différemment par deux équipes.

**Réflexe d'ingénierie.** Pour « interfaces: là où les bons sous-systèmes peuvent devenir un mauvais système », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Une exigence n'est vérifiable ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Une exigence n'est vérifiable ».

### Approfondissement — Gestion des interfaces: la plupart des systèmes échouent aux frontières entre équipes

Une interface décrit ce qui traverse une frontière: énergie, données, fluide, charge mécanique, chaleur, volume, calendrier ou responsabilité opérationnelle. Une interface électrique peut spécifier tension, courant maximal, connecteur, broches, mise à la masse et comportement pendant un défaut. Une interface de données ajoute protocole, débit, format, synchronisation et gestion d'erreur. Une interface mécanique fixe dimensions, rigidité, couples de serrage et charges admissibles.

Le document d'interface n'est utile que s'il possède un propriétaire et une version contrôlée. Si l'équipe A change un connecteur et que l'équipe B continue à fabriquer l'ancienne contrepartie, chaque sous-système peut être parfaitement conforme à son propre dossier et l'ensemble devient inutilisable. Le MBSE, ou *Model-Based Systems Engineering*, peut aider à représenter ces dépendances, mais un modèle numérique ne remplace pas la discipline de décision: qui a le droit de modifier l'interface, qui doit approuver, et comment toutes les équipes apprennent-elles le changement?

### 5. Budgets et marges comme outils de décision

---

Masse, puissance, énergie, débit de données, volume, temps d'équipage et performance sont suivis comme des budgets. La marge n'est pas un pourcentage décoratif: elle absorbe incertitude et maturation. Quand une marge diminue, la décision doit être visible. Les budgets créent un langage commun entre disciplines qui autrement optimiseraient chacune leur propre critère.

**Réflexe d'ingénierie.** Pour « budgets et marges comme outils de décision », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Gestion des interfaces: ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Gestion des interfaces: ».

### 6. MBSE: utiliser un modèle pour relier les produits d'ingénierie

---

Model-Based Systems Engineering ne signifie pas produire de beaux diagrammes. Un modèle utile relie exigences, fonctions, composants, interfaces et preuves de V&V afin qu'un changement laisse une trace. SysML peut formaliser ces relations. Le modèle ne remplace pas le jugement; il réduit surtout les incohérences cachées entre documents indépendants.

**Réflexe d'ingénierie.** Pour « mbse: utiliser un modèle pour relier les produits d'ingénierie », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Gestion des interfaces: ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Gestion des interfaces: ».

## 7. Vérification et validation: deux questions différentes

---

La **vérification** demande si le produit respecte ses exigences: analyse, inspection, démonstration ou essai. La **validation** demande si le produit convient réellement à l'usage et aux attentes de mission. Un système peut être parfaitement vérifié contre de mauvaises exigences et pourtant être inutile. Les deux boucles doivent donc rester visibles depuis le début du projet.

**Réflexe d'ingénierie.** Pour « vérification et validation: deux questions différentes », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Gestion des interfaces: ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Gestion des interfaces: ».

### Approfondissement — Vérification, validation et qualification: trois questions différentes

La vérification demande: « avons-nous construit conformément aux exigences? ». La validation demande: « les exigences décrivent-elles réellement ce dont la mission a besoin? ». On peut donc vérifier parfaitement un mauvais besoin. La qualification ajoute la preuve qu'une conception supporte son environnement avec les marges prévues. Par exemple, une vanne peut respecter son débit nominal sur un banc d'essai, puis échouer après des cycles de **poussière** et de **température**: la fonction a été vérifiée dans un contexte trop faible, pas qualifiée pour Mars.

Une stratégie de V&V, abréviation de *Verification and Validation*, doit donc monter du composant au système: tests unitaires, sous-ensembles, matériel intégré, simulation, essais environnementaux puis démonstrations opérationnelles. Certaines fonctions ne peuvent être validées qu'avec des humains représentatifs. Une alarme peut être techniquement audible mais impossible à distinguer parmi cinquante autres. C'est pourquoi l'ingénierie système doit relier exigences matérielles, logiciel, opérations et **facteurs humains** au lieu de valider chaque discipline dans son propre couloir.

## 8. Matrice de vérification et traçabilité

---

Pour chaque exigence importante, on identifie une méthode, un niveau, une **configuration** d'essai, un responsable et un résultat attendu. Cette **matrice** évite de découvrir trop tard qu'une exigence est impossible à prouver. La traçabilité permet aussi de répondre à la question inverse: si un test échoue, quelles exigences, fonctions et décisions sont affectées?

**Réflexe d'ingénierie.** Pour « matrice de vérification et traçabilité », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Vérification, validation et qualification ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Vérification, validation et qualification ».

## 9. Configuration et changement

---

Un système spatial évolue. La gestion de configuration établit ce qui constitue la référence à un instant donné: exigences, logiciel, dessins, paramètres et procédures. Une modification doit être évaluée pour ses effets transversaux. Sans baseline et contrôle des changements, une correction locale peut réintroduire un défaut ailleurs — exactement le problème que l'anti-régression cherche à empêcher.

**Réflexe d'ingénierie.** Pour « configuration et changement », noter d'abord les entrées, les sorties, les unités et la plage de validité. Puis construire un cas nominal et un cas dégradé — « Approfondissement — Vérification, validation et qualification ». Ce double calcul empêche de confondre une relation mathématique correcte avec une architecture réellement exploitable sur Mars — « Approfondissement — Vérification, validation et qualification ».

## Exemple calculé pas à pas

---

La méthode de travail est toujours la même: écrire ce que représente chaque symbole, convertir toutes les unités vers un système cohérent, effectuer l'opération, puis traduire le résultat en phrase — « Approfondissement — Vérification, validation et qualification ». Enfin, faire un contrôle d'**ordre de grandeur**. Pour « Approfondissement — Vérification, validation et qualification », si le résultat change de facteur mille lorsqu'on passe de millimètres à mètres, la **conversion** doit être visible dans le calcul.

## Exercice progressif

---

1. Choisissez un cas élémentaire de « exigence », puis dressez les données utiles à « Ingénierie système, MBSE & V&V », leur origine et leurs unités avant de calculer.
2. Calculer le résultat nominal sans marge.
3. Modifier le paramètre le plus incertain de  $\pm 20\%$  et comparer — « Approfondissement — Vérification, validation et qualification ».
4. Ajouter une panne crédible et expliquer quel indicateur permet de la détecter — « Approfondissement — Vérification, validation et qualification ».

## Solution raisonnée

Pour « Ingénierie système, MBSE & V&V », le résultat final doit être accompagné de son unité, de l'hypothèse qui le domine et de ce qu'il change pour « exigence ». Elle montre les conversions, la logique de la formule, la sensibilité et la décision — module 21 · Solution raisonnée. Si deux hypothèses différentes conduisent à la même décision opérationnelle, la conception est relativement robuste à cette incertitude. Dans le corrigé de « Ingénierie système, MBSE & V&V », une variation capable d'inverser la conclusion sur « exigence » devient une priorité de mesure ou de marge.

## Mini-projet de validation

Construisez un dossier de validation consacré à « Ingénierie système, MBSE & V&V »: il doit rendre vérifiable au moins un choix concernant « exigence » et expliciter le cas qui ferait rejeter ce choix. Le document doit contenir: besoin, hypothèses, schéma fonctionnel, calcul manuel, vérification par un second calcul ou une simulation, incertitudes, panne injectée, critères de décision et trois références primaires — « Solution raisonnée ». Le mini-projet sur « Ingénierie système, MBSE & V&V » est recevable si un autre équipier peut reproduire la vérification de « exigence » et retrouver la même décision à partir des mêmes données.

## Erreurs classiques à détecter

- Attribuer une cause à « exigence » sans vérifier un indice indépendant du même phénomène.
- Utiliser une valeur de « exigence » sans indiquer son unité, son origine ou sa plage de validité.
- Conclure sur « interface » à partir du cas nominal alors qu'un état dégradé change la décision.
- Dans « Ingénierie système, MBSE & V&V », ne laissez jamais une précision numérique masquer l'hypothèse qui commande réellement le résultat.

## Sources primaires et passerelles

- [NASA — Systems Engineering Handbook](#)
- [NASA — Fundamentals of Systems Engineering](#)

### Repères lexicaux du module. état dégradé

- [www.nasa.gov](#) — documentation institutionnelle
- [standards.nasa.gov](#) — documentation institutionnelle
- [Space Academy](#) — parcours complet

## Matrice de parité FR ↔ EN

| FR                            | EN                                   |
|-------------------------------|--------------------------------------|
| Couverture de vérification    | Verification coverage                |
| Marge absolue de masse        | Absolute mass margin                 |
| Marge relative à l'allocation | Margin as a percentage of allocation |
| Couverture de traçabilité     | Traceability coverage                |
| Fermeture des interfaces      | Interface closure                    |