

Spacecraft: learning to think in systems

Think in systems, interfaces and margins

Starting question — How do subsystem budgets and margins prevent a spacecraft from becoming impossible to integrate?

Intuition. A subsystem can be locally excellent and still break the mission through mass, power, thermal, data or interface conflicts. Engineering therefore manages totals, interfaces and margins continuously.

- Explain the governing physical idea before calculating.
- Name every symbol and unit used in the key relation.
- Check the result with an independent inverse, bound or order-of-magnitude test.

Spacecraft: think in services and interfaces

Budgets connect the subsystems



Delta-Sierra · Bible Mars

An interplanetary spacecraft is not a shopping list of equipment. It is a system of systems: power, thermal control, structures, avionics, propulsion, communications, life support and crew share mass, volume, heat, power, data and failure modes. This module teaches architecture through budgets and interfaces.

1. Start with functions, not hardware names

Define functions first: generate and distribute power, control temperature, know vehicle state, communicate, command actuators, store resources, protect crew and survive failures. Several technologies can satisfy one function; separating need from product keeps the architecture flexible.

2. Mass budget closes the first loop

Exercise A

Vehicle limit 50 t: structure 10, dry propulsion 5, power 4, thermal 2, avionics/comms 2, habitat/crew 12, consumables 8. Find remaining margin.

Total = 43 t; absolute margin = 7 t. That is 14% of the 50 t capacity, or about 16.3% of the 43 t requirement under the relative-margin definition used below. That does not close volume, centre of gravity or propulsion performance.

Decompose functions before choosing hardware

A spacecraft can be described by a small set of functions: maintain trajectory, produce and distribute energy, control temperature, communicate, estimate its state, protect crew or payload and execute mode changes. This functional decomposition prevents the design from starting as a shopping list. “Know attitude,” for example, can involve star trackers, inertial sensors and estimation software; “change attitude” can use wheels, thrusters or other actuators. The need is first stated as performance, availability and accuracy, then hardware is selected. This method also reveals functions that depend on the same resource and therefore exposes common-cause failures that a component list can hide.

3. Power budgets need average, peak and energy

Power in watts is energy rate. A 5 kW load for 8 h uses 40 kWh. Battery design must also respect depth of discharge, efficiency, temperature and peak power.

Exercise B

Critical load 5 kW for 8 h, 80% usable battery fraction and 90% overall efficiency.

Required nominal capacity = $40 \div (0.80 \times 0.90) \approx 55.6$ kWh, before ageing and thermal margin.

4. Electrical power becomes heat

Computers, pumps, lights and electronics dissipate heat. In vacuum that heat eventually has to radiate away. Power and thermal design are coupled; higher equipment power can require more radiator area and mass.

Close power and energy together

A power budget in watts does not size a battery by itself; energy matters as well. If an 800 W load must operate for 3 h with no generation, ideal energy demand is $E = Pt = 800 \times 3 = 2,400$ Wh, or 2.4 kWh. If only 80% of battery capacity is allowed to be used, nominal capacity must exceed $2.4/0.8 = 3.0$ kWh before ageing and margin are added. That storage has mass and produces heat through losses. A credible architecture therefore closes average power, peaks, mode duration, storage, converter efficiency and thermal rejection together. Margin is useful when it corresponds to an identified uncertainty rather than several uncoordinated percentages stacked on the same quantity.

5. Avionics must know, decide and recover

Sensors measure, computers estimate state, software decides and actuators change the vehicle. FDIR—Fault Detection, Isolation and Recovery—detects abnormal behaviour, isolates faults and moves the system into a recoverable configuration. Safe mode is not “turn everything off”; it preserves a stable combination of power, thermal control, attitude and communications.

Modes make architecture visible

A spacecraft does not use every subsystem in the same way during cruise, trajectory correction, crew sleep, high-rate communication or survival. A mode table states which functions are active, which resources they consume, which transitions are allowed and what evidence permits exit. This immediately exposes interfaces: a safe mode that removes a data bus may also remove the sensor needed to diagnose the original event. FDIR therefore has to preserve a minimum command and telemetry path. Recovery should not mean rebooting everything at once. It should move through known configurations, confirming critical resources as the vehicle returns to a more capable state.

6. Modes expose interfaces

Nominal cruise, manoeuvre, occultation, solar event, power loss, safe mode and Mars approach activate different priorities. A mode $\times$ function matrix reveals dependencies: high-gain communications may conflict with manoeuvre attitude; radiator pointing can constrain orientation; propulsion can disturb sensors.

7. Common causes defeat naive redundancy

Two identical computers do not protect against one shared software defect. Two pumps on one bus do not protect against bus loss. Reliability analysis searches for shared power, cooling, software, reference sensors, valves, connectors and procedures.

Modes are also power and thermal configurations

A spacecraft mode is not merely a software label. It determines which loads are powered, which heaters operate, which antennas point, which sensors remain awake and how much heat is generated. A survival mode that cuts 1 kW of payload power may protect the battery but also reduce internal heat and change thermal balance. Conversely, high-rate communications can increase transmitter power and require a pointing attitude that changes solar input. Mode design therefore needs a table of power, energy, thermal and data consequences. Transition criteria should be based on measured state—battery energy, temperature, fault status or attitude—not just elapsed time. This makes safe-mode behaviour predictable during real anomalies.

8. Verification must prove interfaces

Testing one box does not prove the spacecraft. Integration testing checks data, electrical, thermal, mechanical and sequencing interfaces. Hardware-in-the-loop testing puts real flight computers in simulated environments. Degraded modes and recovery deserve tests, not only nominal operation.

9. Read technology surveys within their scope

NASA's 2026 Small Spacecraft report is useful for current component families in power, GNC, thermal, avionics and communications, but its scope is explicitly SmallSat. It does not qualify a crewed Mars spacecraft. A serious course separates component state-of-the-art from system-level evidence.

Verification proves requirements; validation proves usefulness

Verification asks whether a defined requirement has been met: voltage range, battery capacity, pointing accuracy, structural strength or software response. Validation asks whether the complete requirement set and solution actually satisfy the mission need. A system can pass every formal verification and still be operationally wrong. Mars-spacecraft evidence therefore comes from multiple levels: component tests, software tests, subsystem benches, interface tests, environmental campaigns and integrated scenarios. Because no Earth facility can reproduce a multi-year Mars mission in full, analysis and simulation fill gaps. Every claim should state which part is directly demonstrated and which part remains extrapolated.

10. Check yourself

- Function versus component?
- Repeat exercise A after adding 3 t shielding.
- Why are 5 kW and 40 kWh different?
- Give one common-cause failure.
- What must safe mode preserve?

11. Build an interface matrix

Use five subsystems: power, thermal, avionics, propulsion and communications. At every row-column intersection write what crosses the interface: watts, heat, data, mechanical torque, fluid, timing or command. The matrix exposes hidden dependencies. Communications consumes power, rejects heat, constrains attitude and exchanges data with avionics.

12. First radiator estimate with Stefan-Boltzmann

An ideal radiator follows $P = \epsilon \sigma A T^4$. P is watts, ϵ emissivity, $\sigma \approx 5.670 \times 10^{-8} \text{ W/m}^2/\text{K}^4$, A area in m^2 and T kelvins. Real geometry adds solar and planetary fluxes, but the equation shows the strong fourth-power temperature effect.

Exercise C

A spacecraft must reject 5,000 W through an ideal radiator with emissivity $\varepsilon = 0.85$ at $T = 300$ K. Using $P = \varepsilon\sigma AT^4$ with $\sigma = 5.670374419 \times 10^{-8} \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-4}$, calculate the ideal radiator area. Then name two real-design effects that would require additional area or margin.

Detailed correction — Exercise C

Radiative flux. $\varepsilon\sigma T^4 = 0.85 \times 5.670374419 \times 10^{-8} \times 300^4 \approx 390 \text{ W/m}^2$.

Area. $A = 5,000 \text{ W} \div 390 \text{ W/m}^2 \approx 12.8 \text{ m}^2$.

Engineering margin. View factors, radiator orientation, contamination, degradation, operating temperature variation and absorbed external heat can all increase required area.

An interface matrix exposes hidden dependence

For every subsystem pair, the interface matrix records what crosses the boundary: power, data, heat, fluid, mechanical load, radio-frequency energy or command authority. Each interface has a normal envelope and failure cases. A computer may receive correct voltage yet lose a coherent time reference; a healthy radiator can become ineffective if coolant flow no longer carries heat to it. The matrix also assigns ownership of each requirement and a verification method. As the design evolves, this prevents a local modification from silently consuming another subsystem's margin. Interface discipline is therefore one of the most practical tools for preventing common-cause failures.

13. Data rate, latency and criticality are different

A camera can produce high data volume without millisecond criticality. An IMU produces less data with strict timing. Avionics networks need priority, saturation handling and failure behaviour; redundant buses do not help if one software fault floods both.

14. Configuration means knowing exactly what flies

Long-duration spacecraft may receive updates. Configuration ties hardware, firmware, software, parameters, calibration tables and procedures together. A safe update process includes verification and rollback.

Data rate, latency and criticality must be separated

A high data rate does not guarantee useful control. Telemetry can arrive with delay, be incomplete or contain low-priority detail while a small critical state vector is missing. Communications design therefore classifies data by operational value. A fault message may need only a few bytes but must include time and configuration context; a science image can be megabytes and tolerate delay. Buffers are sized in bits or bytes from generation rate and outage duration. For example, 2 Mbit/s generated during a 30 min communications gap requires $2 \times 10^6 \times 1,800 = 3.6 \times 10^9$ bits, about 450 MB before protocol overhead and margin. Storage and link scheduling are therefore part of spacecraft architecture.

15. FDIR: detect, isolate, recover

Detection identifies inconsistency, isolation localises the likely fault, and recovery reconfigures the vehicle around it. Response time depends on the hazard: some loops need milliseconds, other problems allow minutes.

Exercise D — false consensus

Two identical computers agree while an independent sensor disagrees. Can the sensor be declared wrong?

No. The computers can share software, input or configuration errors. Seek independent evidence: another sensor, physical model, active test, history or manual measurement.

16. Verification matrix: requirement → evidence

Every requirement needs a verification method—analysis, inspection, test or demonstration—and explicit success criteria. Validation asks a different question: even if requirements are met, does the system satisfy the real mission need?

FDIR recovery has to be demonstrated

FDIR means Fault Detection, Isolation and Recovery. Detection recognises that the observed state is inconsistent with expectations; isolation identifies the responsible function or chain; recovery places the spacecraft in a safe or useful condition. The three steps need not reach perfect certainty at the same instant. A fault can be detected quickly and isolated later if the interim mode protects critical resources. Tests should inject realistic faults—stuck sensors, inconsistent data, bus loss, slow actuators—and verify not only that an alarm appears but that the final vehicle state is controlled. An FDIR design that launches an opaque cascade of automatic actions can be more dangerous than the initial fault.

17. Mini-project — review architecture under failure

1. Draw power, thermal, avionics, communications and propulsion services.
2. Add interfaces.
3. Remove the primary power bus.
4. List functions required for 30 minutes.
5. Calculate emergency energy.
6. Check thermal consequences.
7. Define minimum telemetry.
8. Describe return to service.

18. Next level: system models

Model-Based Systems Engineering tools can organise requirements, functions and interfaces when complexity grows. They do not replace reasoning. A sophisticated model with one wrong unit or missing dependency remains wrong, so Space Academy builds traceable hand calculations before tool complexity.

Mini-project: close a failure across subsystem boundaries

Choose one failure such as loss of a main power converter. Trace the immediate electrical effect, the loads that disappear, the thermal consequence of reduced dissipation, the telemetry still available and the action that attitude control or crew procedures must take. Then identify the independent path needed for recovery and the configuration evidence required before reconnecting loads. Quantify at least one resource, such as battery energy during the degraded interval. The exercise is successful when the recovery path crosses several subsystem boundaries without relying on the failed resource. This is the difference between drawing redundant boxes and demonstrating fault containment in a real architecture.

19. Mass, power and heat margins interact

Adding a processor may look like a small avionics change, but it draws electrical power, produces heat, needs data connections and may require shielding or spare capacity. The correct change request therefore updates several budgets. Systems engineering exists partly to prevent local improvements from breaking global margins.

Thermal rejection and configuration are first-class architecture concerns

Electrical power, internal dissipation and radiator capability must be closed together. A radiator is often estimated first with Stefan–Boltzmann radiation, $P = \epsilon \sigma A (T^4 - T_{env}^4)$, where P is radiated power in watts, ϵ emissivity without unit, σ the Stefan–Boltzmann constant, A area in m^2 and T absolute temperature in kelvins. The equation is only a first estimate because view factors, coatings, conductive paths and changing solar geometry matter. Configuration control is equally physical: a radiator model, heater setting or flight-software limit is valid only for the hardware and software version it describes. Long-duration spacecraft therefore need traceable configuration records so that an operational limit is never applied to the wrong vehicle state.

20. Fault containment is an architectural property

A fault should ideally remain inside a bounded region. Electrical protection prevents one short from collapsing the whole bus; network segmentation prevents one node from saturating all traffic; software partitioning reduces propagation between functions. Containment often matters more than simply adding duplicate hardware.

21. Crew interaction is another interface

Human operators need displays, procedures and controls that match the automation state. If the spacecraft changes mode autonomously, crew must know why, what authority remains, and which action is safe. A technically correct FDIR response can still create risk if humans misunderstand the configuration.

22. Architecture review exercise



A Martian greenhouse becomes a subsystem consuming power, water and thermal control while producing biomass, humidity and maintenance workload.

Choose one requirement such as “survive eight hours without primary generation.” Derive the energy reserve, loads to shed, thermal consequences, communication minimum and recovery test. This converts a vague resilience statement into a cross-subsystem evidence chain.

Architecture review: ask what is not represented in the diagram



Fluid loops combine pumps, filters, tanks, sensors and automation; their availability depends as much on maintenance as on nominal efficiency.

A block diagram is a model and therefore omits things. Reviewers should ask whether it shows timing, shared software, common cooling, ground tools, crew actions, calibration, configuration and maintenance access. A line labelled “data” may hide multiple protocols and clocks; a box labelled “power” may hide a single common converter. The most important review questions often concern those omitted dependencies. For each life- or mission-critical function, identify the first failure that removes it, the second failure that becomes dangerous after degradation, and the observation that proves recovery. This turns an architecture review from a presentation of boxes into a search for hidden coupling and unclosed evidence.

Engineering studio — close mass, power and margin

The exercise spacecraft has a 60 t target mass, 18 kW of available electrical generation in the studied mode and 14 kW of committed loads. A new computing unit adds 0.4 t and 1.2 kW. Power margin falls from 4.0 kW to 2.8 kW, a 30% reduction. The calculation forces two budgets to be tracked at once: mass in tonnes and instantaneous power in kilowatts; neither quantity can simply be exchanged for the other.

In degraded mode, a converter adds another 0.8 kW of loss. The student selects loads that can be shed without losing safe navigation, thermal control or communications, then identifies secondary effects: additional waste heat, longer battery duty and potentially extra radiator or wiring mass. The capstone therefore treats the spacecraft as a coupled system.

For the spacecraft module, the closing review treats a new subsystem as a coupled change request. Added mass changes propulsion demand; added electrical load changes generation, storage and rejected heat; additional software changes verification workload. The student must identify at least one second-order effect outside the subsystem that requested the change and show which system-level margin records it.

Coupled budget — one extra kilowatt becomes mass and heat

Assume a payload needs 1 kW continuously. If conversion efficiency is 90%, the upstream electrical system must supply about 1.11 kW and roughly 0.111 kW appears as conversion loss. Over a 24-hour day that is about 2.67 kWh of waste energy that the thermal system must reject or store. The calculation is simple, but it exposes a system rule: an electrical change is simultaneously a thermal and mass change.

Now add a battery requirement for two hours of survival without generation. The battery must cover the useful load plus conversion losses, and its own mass affects structure and manoeuvre performance. A spacecraft budget is therefore a network of coupled quantities. The goal of the exercise is not to find one “right” battery; it is to trace how one requirement propagates into several subsystems.

Calculation laboratory — certified formula reasoning

This laboratory adds the zero-prerequisite reasoning chain used by the certified French Academy: read the formula, define every symbol and unit, calculate step by step, then verify the result independently.

Spacecraft architecture: canonical mini-lessons

These mini-lessons form the module’s canonical quantitative reference. Numerical cases are teaching examples and must remain distinct from mission data.

Relative capacity margin

$$m_{\text{rel}} = (C - B) / B$$

1 — Concrete question

What does “ $m_{\text{rel}} = (C - B) / B$ ” compute in the context of “Relative capacity margin”?

2 — Intuition without symbols

Relative margin compares excess capacity with the requirement baseline, not total capacity.

3 — Quantities

m_{rel} : relative margin; C: capacity; B: requirement.

4 — Formula

$$m_{\text{rel}} = (C - B) / B$$

5 — Read aloud

Read “ $m_{\text{rel}} = (C - B) / B$ ” by naming every operation explicitly.

6 — Symbols and meaning

m_{rel} : relative margin; C: capacity; B: requirement.

7 — Pronunciation

Say subscripts, exponents, Greek letters and operations explicitly so another person can reconstruct the relation without seeing the page.

8 — Units

C and B in the same unit; m_{rel} dimensionless or percent.

9 — Convention

Keep the same frame, sign convention, time basis and system boundary throughout the calculation.

10 — Why this operation

First compute excess $C-B$, then divide by requirement B.

11 — Assumptions

Same scope and definition for capacity and requirement.

12 — Unit check

C and B in the same unit; m_{rel} dimensionless or percent. Verify that units reduce to the announced output quantity.

13 — Numerical case

With $C = 50 \text{ t}$ and $B = 43 \text{ t}$, $m_{\text{rel}} = 7/43 = 0.1628 = 16.3\%$.

14 — Why the calculation works

First compute excess $C-B$, then divide by requirement B.

15 — Algebraic check

$B \times (1 + m_{\text{rel}})$ must recover about 50 t.

16 — Mental estimate

7 t over a 43 t requirement is a little over one sixth, about 16%.

17 — Interpretation

Margin is relative to the chosen baseline; changing denominator changes the percentage.

18 — What the result does not prove

This result proves neither universal validity of the model nor quality of the inputs; it applies only to the stated scenario and assumptions.

19 — Sensitivity

At fixed requirement, higher C increases margin linearly; at fixed capacity, higher requirement reduces it.

20 — Guided and autonomous exercises

Guided exercise. C = 20,000 kg and B = 17,000 kg.

Detailed guided correction — open after trying

$m_{rel} = 3,000/17,000 = 0.1765 = 17.6\%$.

Autonomous exercise. C = 12,000 kg and B = 10,000 kg.

Autonomous correction — open after trying

$m_{rel} = 2,000/10,000 = 0.20 = 20\%$.

21 — Mission decision

State margin baseline explicitly before any mass or capacity decision.

Energy consumed by a load

$E = P \times t$

1 — Concrete question

What does “ $E = P \times t$ ” compute in the context of “Energy consumed by a load”?

2 — Intuition without symbols

Power sustained over time creates energy demand equal to their product.

3 — Quantities

E: energy; P: power; t: duration.

4 — Formula

$E = P \times t$

5 — Read aloud

Read “ $E = P \times t$ ” by naming every operation explicitly.

6 — Symbols and meaning

E: energy; P: power; t: duration.

7 — Pronunciation

Say subscripts, exponents, Greek letters and operations explicitly so another person can reconstruct the relation without seeing the page.

8 — Units

$W \times h = Wh$ or $W \times s = J$, depending on the chosen basis.

9 — Convention

Keep the same frame, sign convention, time basis and system boundary throughout the calculation.

10 — Why this operation

Power is energy flow; multiplying by time gives total energy.

11 — Assumptions

Representative average power and consistent duration.

12 — Unit check

$W \times h = Wh$ or $W \times s = J$, depending on the chosen basis. Verify that units reduce to the announced output quantity.

13 — Numerical case

An 800 W load for 3 h requires $E = 800 \times 3 = 2,400 Wh = 2.4 kWh$.

14 — Why the calculation works

Power is energy flow; multiplying by time gives total energy.

15 — Algebraic check

E/t must recover 800 W.

16 — Mental estimate

0.8 kW for 3 h gives 2.4 kWh.

17 — Interpretation

The result supports storage and generation sizing.

18 — What the result does not prove

This result proves neither universal validity of the model nor quality of the inputs; it applies only to the stated scenario and assumptions.

19 — Sensitivity

E varies linearly with P and t.

20 — Guided and autonomous exercises

Guided exercise. P = 1.2 kW for 5 h.

Detailed guided correction — open after trying

$E = 1.2 \times 5 = 6.0$ kWh.

Autonomous exercise. P = 0.5 kW for 8 h.

Autonomous correction — open after trying

$E = 0.5 \times 8 = 4.0$ kWh.

21 — Mission decision

Check energy across no-generation phases, not only peak power.

Nominal battery capacity

$C_{\text{nom}} = E_{\text{useful}} / (\text{DoD} \times \eta)$

1 — Concrete question

What does " $C_{\text{nom}} = E_{\text{useful}} / (\text{DoD} \times \eta)$ " compute in the context of "Nominal battery capacity"?

2 — Intuition without symbols

Battery must be larger than useful energy demand because full depth is not used and losses exist.

3 — Quantities

C_{nom} : nominal capacity; E_{useful} : useful energy; DoD: usable depth of discharge; η : overall efficiency.

4 — Formula

$C_{\text{nom}} = E_{\text{useful}} / (\text{DoD} \times \eta)$

5 — Read aloud

Read " $C_{\text{nom}} = E_{\text{useful}} / (\text{DoD} \times \eta)$ " by naming every operation explicitly.

6 — Symbols and meaning

C_{nom} : nominal capacity; E_{useful} : useful energy; DoD: usable depth of discharge; η : overall efficiency.

7 — Pronunciation

Say subscripts, exponents, Greek letters and operations explicitly so another person can reconstruct the relation without seeing the page.

8 — Units

C_{nom} and E_{useful} in the same energy unit; DoD and η dimensionless.

9 — Convention

Keep the same frame, sign convention, time basis and system boundary throughout the calculation.

10 — Why this operation

Actually available energy is $C_{\text{nom}} \times \text{DoD} \times \eta$; invert this relation to size C_{nom} .

11 — Assumptions

DoD and efficiency representative of the mission conditions considered.

12 — Unit check

C_{nom} and E_{useful} in the same energy unit; DoD and η dimensionless. Verify that units reduce to the announced output quantity.

13 — Numerical case

For $E_{\text{useful}} = 40$ kWh, DoD = 0.80 and $\eta = 0.90$, $C_{\text{nom}} = 40/0.72 = 55.56$ kWh.

14 — Why the calculation works

Actually available energy is $C_{\text{nom}} \times \text{DoD} \times \eta$; invert this relation to size C_{nom} .

15 — Algebraic check

$55.56 \times 0.80 \times 0.90$ must recover about 40 kWh.

16 — Mental estimate

Dividing by 0.72 raises capacity by about 39% above useful energy.

17 — Interpretation

The result is an energy minimum before ageing, temperature and system margin.

18 — What the result does not prove

This result proves neither universal validity of the model nor quality of the inputs; it applies only to the stated scenario and assumptions.

19 — Sensitivity

Lower DoD or η increases required nominal capacity.

20 — Guided and autonomous exercises

Guided exercise. $E_{\text{useful}} = 24$ kWh, DoD = 0.80, $\eta = 0.90$.

Detailed guided correction — open after trying

$C_{\text{nom}} = 24/0.72 = 33.33$ kWh.

Autonomous exercise. $E_{\text{useful}} = 18$ kWh, DoD = 0.75, $\eta = 0.85$.

Autonomous correction — open after trying

$C_{\text{nom}} = 18/0.6375 = 28.24$ kWh.

21 — Mission decision

Add ageing, temperature and contingency before freezing real battery size.

Ideal radiator area

$$A = P / [\varepsilon \sigma (T^4 - T_{\text{env}}^4)]$$

1 — Concrete question

What does “ $A = P / [\varepsilon \sigma (T^4 - T_{\text{env}}^4)]$ ” compute in the context of “Ideal radiator area”?

2 — Intuition without symbols

A radiator must provide enough area to radiate thermal power at available temperature and emissivity.

3 — Quantities

A: area; P: thermal power; ε : emissivity; σ : Stefan–Boltzmann constant; T: radiator temperature; T_{env} : radiative environment temperature.

4 — Formula

$$A = P / [\varepsilon \sigma (T^4 - T_{\text{env}}^4)]$$

5 — Read aloud

Read “ $A = P / [\varepsilon \sigma (T^4 - T_{\text{env}}^4)]$ ” by naming every operation explicitly.

6 — Symbols and meaning

A: area; P: thermal power; ε : emissivity; σ : Stefan–Boltzmann constant; T: radiator temperature; T_{env} : radiative environment temperature.

7 — Pronunciation

Say subscripts, exponents, Greek letters and operations explicitly so another person can reconstruct the relation without seeing the page.

8 — Units

A in m²; P in W; temperatures in K; $\sigma = 5.670374419 \times 10^{-8}$ W/m²/K⁴.

9 — Convention

Keep the same frame, sign convention, time basis and system boundary throughout the calculation.

10 — Why this operation

Stefan–Boltzmann law gives net radiative flux; dividing rejected power by that flux gives area.

11 — Assumptions

Ideal radiative view and uniform temperatures; conductive/convective exchanges excluded.

12 — Unit check

A in m²; P in W; temperatures in K; $\sigma = 5.670374419 \times 10^{-8}$ W/m²/K⁴. Verify that units reduce to the announced output quantity.

13 — Numerical case

For P = 5,000 W, $\epsilon = 0.85$, T = 300 K and T_{env} ≈ 3 K, A ≈ 12.81 m².

14 — Why the calculation works

Stefan–Boltzmann law gives net radiative flux; dividing rejected power by that flux gives area.

15 — Algebraic check

$A \times \epsilon \sigma (T^4 - T_{\text{env}}^4)$ must recover about 5,000 W.

16 — Mental estimate

At 300 K and ϵ near 1, flux is a few hundred W/m²; several kW therefore require several m².

17 — Interpretation

The result is ideal area before view factors, degradation, shadowing and mechanical constraints.

18 — What the result does not prove

This result proves neither universal validity of the model nor quality of the inputs; it applies only to the stated scenario and assumptions.

19 — Sensitivity

A falls strongly as T rises because radiation scales with T⁴.

20 — Guided and autonomous exercises

Guided exercise. P = 3,000 W, $\epsilon = 0.90$, T = 290 K, T_{env} = 3 K.

Detailed guided correction — open after trying

Net flux is about 360.9 W/m²; $A \approx 3,000/360.9 = 8.31$ m².

Autonomous exercise. P = 8,000 W, $\epsilon = 0.80$, T = 310 K, T_{env} = 3 K.

Autonomous correction — open after trying

Net flux is about 418.9 W/m²; $A \approx 8,000/418.9 = 19.10$ m².

21 — Mission decision

Size thermal system with real margins and hot/cold cases, not ideal area alone.

Mission reasoning lab — close a spacecraft mass budget

Scenario. A spacecraft has a maximum allowable launch mass of 20,000 kg. Current subsystem estimates total 17,000 kg. The simple absolute margin is therefore 3,000 kg and the relative margin against the 17,000-kg requirement basis is $3,000/17,000 \approx 17.6\%$. The exact convention used by a program may differ; what matters is that capacity, requirement and denominator are explicitly defined.

1. A budget is a controlled interface

Mass, power, thermal dissipation, data rate and volume budgets all perform the same systems-engineering function: they connect local subsystem decisions to a system-level limit. If one team silently changes an assumption, every dependent budget can become inconsistent.

2. Do not mix current estimate and allocated allowance

A subsystem may currently weigh 900 kg but hold a 1,050-kg allocation. Those are different numbers with different purposes. The current best estimate describes today's design; the allocation reserves system capacity. A margin report must state which one enters the sum.

3. Reconstruct capacity from margin

If requirement is 17,000 kg and relative margin is 17.6%, the inverse relation gives capacity $\approx 17,000 \times 1.176 \approx 19,992$ kg, close to 20,000 kg after rounding. This inverse check is useful because spreadsheet errors often arise from dividing by capacity instead of requirement or from applying a percentage twice.

4. Negative margin is information, not an accounting trick

If estimates grow to 20,500 kg against a 20,000-kg limit, the system is 500 kg over. Calling the margin “-2.4%” does not solve the problem. Engineering must remove mass, change architecture, change the launch vehicle or revise the requirement through a controlled decision. Hidden negative margin becomes a late integration crisis.

5. Coupled budgets

Reducing mass can increase power or thermal demand; adding insulation can increase mass but reduce heater power. Systems engineering therefore examines trades across budgets rather than minimizing one resource in isolation. A locally lighter subsystem is not automatically better if it pushes another critical resource over its limit.

6. Margin policy should tighten with maturity

Early concepts carry larger uncertainty and therefore need larger reserves. As design, test evidence and supplier data mature, uncertainty should shrink. A margin that disappears without corresponding evidence is a warning sign; a margin that never changes despite maturation may indicate that the accounting is disconnected from engineering reality.

Decision check

Before accepting a budget, verify the system boundary, unit, reference configuration, maturity date, allocation versus estimate, explicit reserves and the sign convention for margin. Then sum independently. A budget is credible only when another engineer can reproduce it from the same configuration data.

Configuration control and margin discipline

One configuration at a time. Budgets are meaningful only when every contributor refers to the same spacecraft configuration. Mixing a new solar-array mass with an old battery allocation or a pre-change thermal load creates a total that never existed physically. Each budget release should identify configuration and date.

Interfaces consume resources. Harness, adapters, structural joints, software overhead, plumbing and integration hardware often fall between subsystem boundaries. If nobody owns them, the system budget becomes optimistic. Interface mass and power should have explicit owners rather than being treated as invisible overhead.

Uncertainty is not the same as margin. Statistical uncertainty describes what is not known; programme margin is a managed reserve. They can be related but should not be merged casually. A design with high uncertainty may need more margin, yet the rationale and confidence level must remain visible.

Trade closure. When a budget exceeds its limit, record the proposed corrective action and its secondary effects. Removing a radiator reduces mass but can violate temperature limits; shrinking batteries saves mass but reduces eclipse endurance. The systems engineer closes the coupled trade, not just the red cell in one spreadsheet.

Final budget check

Every major resource budget should include an explicit reconciliation step after design reviews. When a component changes, the owner updates the local estimate, the subsystem lead confirms interfaces, and the system budget records the

new total and remaining reserve. This prevents a late surprise caused by several individually small changes that were never accumulated together.

For human-rated systems, reserves also have an operational dimension: emergency power, consumables and spare capacity may be intentionally protected from routine use. A budget that spends survival reserve to make the nominal mission close is not genuinely closed.

Zero-prerequisite concepts

These concept cards define the vocabulary that a first-time learner must master before using it operationally.

system

Definition. A system is an organized set of interacting elements that together deliver a defined function within a stated boundary.

Example. A crewed spacecraft is a system containing propulsion, power, avionics, structures, thermal control and life support.

Pitfall. The system boundary must be explicit; otherwise inputs, outputs and responsibilities become ambiguous.

Mental check. If a resource crosses the boundary, it must appear as an interface or external input/output.

Guided exercise — system

Situation to recognize. A system is an organized set of interacting elements that together deliver a defined function within a stated boundary.

Check requested. If a resource crosses the boundary, it must appear as an interface or external input/output.

Error to reject. The system boundary must be explicit; otherwise inputs, outputs and responsibilities become ambiguous.

Reasoned solution

Precise meaning

A system is an organized set of interacting elements that together deliver a defined function within a stated boundary.

Case test

If a resource crosses the boundary, it must appear as an interface or external input/output.

Excluded pitfall

The system boundary must be explicit; otherwise inputs, outputs and responsibilities become ambiguous.

Operational consequence

Use this check before accepting a result in mission design: If a resource crosses the boundary, it must appear as an interface or external input/output.

Quantification

system: use the unit or dimension defined by the physical quantity; if the concept is qualitative, do not invent a numerical unit.

Verification

system: compare the conclusion with the mental check and the stated pitfall.

subsystem

Definition. A subsystem is a coherent part of a larger system that performs a defined subset of functions.

Example. The electrical-power subsystem generates, stores and distributes energy for the spacecraft.

Pitfall. Optimizing a subsystem independently can harm overall mission performance.

Mental check. A subsystem requirement should trace to a system need and interface with neighbouring subsystems.

Guided exercise — subsystem

Situation to recognize. A subsystem is a coherent part of a larger system that performs a defined subset of functions.

Check requested. A subsystem requirement should trace to a system need and interface with neighbouring subsystems.

Error to reject. Optimizing a subsystem independently can harm overall mission performance.

Reasoned solution

Precise meaning

A subsystem is a coherent part of a larger system that performs a defined subset of functions.

Case test

A subsystem requirement should trace to a system need and interface with neighbouring subsystems.

Excluded pitfall

Optimizing a subsystem independently can harm overall mission performance.

Operational consequence

Use this check before accepting a result in mission design: A subsystem requirement should trace to a system need and interface with neighbouring subsystems.

Quantification

subsystem: use the unit or dimension defined by the physical quantity; if the concept is qualitative, do not invent a numerical unit.

Verification

subsystem: compare the conclusion with the mental check and the stated pitfall.

mass budget

Definition. A mass budget allocates and tracks allowable mass across components and subsystems so the total remains compatible with mission limits.

Example. Each subsystem reports current best estimate, growth allowance and margin.

Pitfall. A mass budget is not a one-time spreadsheet; it changes as design matures.

Mental check. The sum of allocated or estimated masses plus appropriate reserves must remain consistent with the system limit.

Guided exercise — mass budget

Situation to recognize. A mass budget allocates and tracks allowable mass across components and subsystems so the total remains compatible with mission limits.

Check requested. The sum of allocated or estimated masses plus appropriate reserves must remain consistent with the system limit.

Error to reject. A mass budget is not a one-time spreadsheet; it changes as design matures.

Reasoned solution

Precise meaning

A mass budget allocates and tracks allowable mass across components and subsystems so the total remains compatible with mission limits.

Case test

The sum of allocated or estimated masses plus appropriate reserves must remain consistent with the system limit.

Excluded pitfall

A mass budget is not a one-time spreadsheet; it changes as design matures.

Operational consequence

Use this check before accepting a result in mission design: The sum of allocated or estimated masses plus appropriate reserves must remain consistent with the system limit.

Quantification

mass budget: use the unit or dimension defined by the physical quantity; if the concept is qualitative, do not invent a numerical unit.

Verification

mass budget: compare the conclusion with the mental check and the stated pitfall.

margin

Definition. A margin is deliberately reserved capacity beyond the current requirement or estimate to absorb uncertainty, growth and design evolution.

Example. Power systems may keep generation or storage margin above expected demand.

Pitfall. An arbitrary percentage applied everywhere can hide poor understanding.

Mental check. At fixed capacity, margin must shrink when the requirement increases.

Guided exercise — margin

Situation to recognize. A margin is deliberately reserved capacity beyond the current requirement or estimate to absorb uncertainty, growth and design evolution.

Check requested. At fixed capacity, margin must shrink when the requirement increases.

Error to reject. An arbitrary percentage applied everywhere can hide poor understanding.

Reasoned solution

Precise meaning

A margin is deliberately reserved capacity beyond the current requirement or estimate to absorb uncertainty, growth and design evolution.

Case test

At fixed capacity, margin must shrink when the requirement increases.

Excluded pitfall

An arbitrary percentage applied everywhere can hide poor understanding.

Operational consequence

Use this check before accepting a result in mission design: At fixed capacity, margin must shrink when the requirement increases.

Quantification

margin: use the unit or dimension defined by the physical quantity; if the concept is qualitative, do not invent a numerical unit.

Verification

margin: compare the conclusion with the mental check and the stated pitfall.

Beginner vocabulary checkpoint

These terms are intentionally interactive: focus or tap a term for its definition, then return to the surrounding calculation or mission decision.

- spacecraft system — Complete set of interacting hardware, software, people and interfaces that deliver mission functions.
- subsystem — Coherent part of a larger system responsible for a defined group of functions.
- interface — Boundary where energy, data, forces, fluids or responsibilities pass between elements.
- mass budget — Allocation and tracking of spacecraft mass across subsystems and reserves.
- power budget — Accounting of electrical generation, storage, conversion and loads.
- thermal control — Methods used to keep hardware and people within allowable temperature limits.
- radiator — Surface designed to reject heat to space mainly by thermal radiation.
- depth of discharge — Fraction of stored battery capacity removed during a discharge cycle.
- avionics — Flight electronics that provide computing, data handling, guidance, navigation and control functions.
- fault tolerance — Ability to continue required operation after specified failures.
- redundancy — Additional independent capability provided to survive a failure.
- single-point failure — One failure that by itself can cause loss of a critical function or mission.
- design margin — Reserved capacity beyond the current requirement or estimate.

- centre of gravity — Point at which the vehicle mass can be treated as concentrated for many force and moment calculations.
- structural load — Force or moment carried by spacecraft structure during launch, manoeuvres or landing.
- consumables — Finite resources used during a mission, such as propellant, oxygen, water or filters.
- harness — Wiring and cabling that distribute electrical power and data.
- configuration control — Discipline that keeps approved hardware, software, interfaces and documentation mutually consistent.

Sources and references

NASA — 2026 State-of-the-Art of Small Spacecraft Technology · NASA Johnson Engineering · NASA/JPL — Mission to Mars Unit.

FR ↔ EN competency matrix

#	Français	English	Status
01	Marge relative capacité/besoin	Relative capacity margin	PASS
02	Énergie consommée par une charge	Energy consumed by a load	PASS
03	Capacité nominale de batterie	Nominal battery capacity	PASS
04	Aire idéale de radiateur	Ideal radiator area	PASS