

Organizational structure

Who owns what

RoboTeam Twente should function as two distinct wings. The STEM and non-STEM.

The non-STEM part is headed by the Team Manager/General Manager and is assisted by human relations and external affairs.

From here there are two different ways of thought, an organization of this scale can structure their STEM wing as either by **specialization** or **subsystem** which is headed by a Technical Manager/Systems Engineer or both.

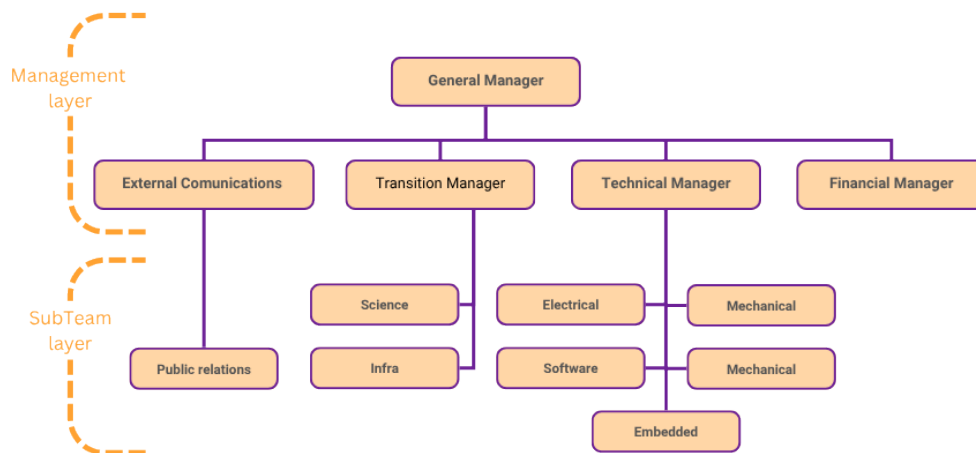
In the past we structured according to **specialization** this entails that you create and work in groups of the same field e.g. Mechanical embedded, electrical, etc.

This wasn't "bad", but it left the gaps between fields unowned, decisions that crossed departments had no clear home and often went unrecorded or unaddressed until it was very late in the process.

The way the team is divided into roles directly shapes how information moves. Where responsibilities are unclear or unowned, decisions get made in conversation and then lost.

This is not a description of how things happen to work. It is a statement of how the team is structured and how information should flow through it, and it exists to remove the ambiguity that caused information to fall between departments.

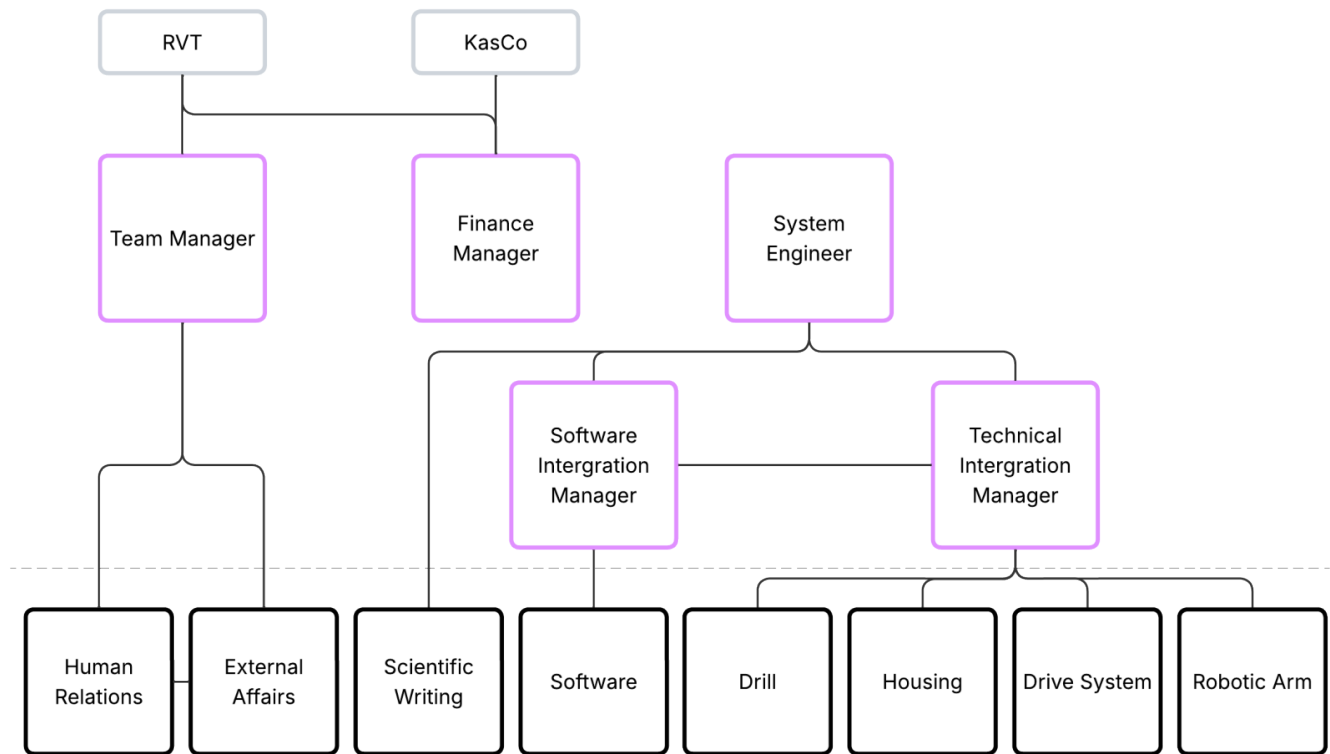
The current structure (2025-2026 specialization)



Historically the team is organized by **specialization**, software, embedded, mechanical, electrical, control, each department led by a lead who reports to the technical manager.

This works for managing people within a department. It does not work for the parts of the robot that **cross** departments. The clearest example: GNSS, multiple teams were working on while it had already been decided that it would not be used as it did not fit within the scope of the competition. No single role owned that interface, so no one was accountable for documenting the current decisions.

The proposed structure (subsystem)



The other option is to structure according to **subsystems**, hereby you divide teams from different specializations in teams according to functionality of the project. For us that would be a Mars rover so below here is an example of what I imagine an effective RoboTeam structure would be. Of course keep in mind the **subsystems** under the Technical Integration Manager are just a rough outline of the idea, these systems can be broken down as specific or as broad as the size of the organization allows and the scope of that years project calls for. The proposed structure is not "purely" subsystem-based. Technical subsystems (drill, housing, drive, arm) are organized by function under the Technical Integration Manager, while software remains a single department under the Software Integration Manager because its work spans all subsystems simultaneously. This hybrid is deliberate and by having a clear line of communication between the software and technical integration managers it should avoid anything falling between the cracks or for one scope to grow disproportionately to the other.

Explanation of managerial positions

Team Manager

- **Owns:** [*the interface register between software and embedded — every agreed dependency, in one documented place*].
- **Why this role exists:** under the current structure, interface changes are communicated ad hoc between two leads, which the interviews link directly to [*late integration failures — cite App. 1.5*]. A single owner means every interface change has one documented point of record.
- **Who fills it:** [*not a new hire — reallocated from an existing member's time; state whose and how many hours*].

Financial Manager

- **Owns:** [*the interface register between software and embedded — every agreed dependency, in one documented place*].
- **Why this role exists:** under the current structure, interface changes are communicated ad hoc between two leads, which the interviews link directly to [*late integration failures — cite App. 1.5*]. A single owner means every interface change has one documented point of record.
- **Who fills it:** [*not a new hire — reallocated from an existing member's time; state whose and how many hours*].

System Engineer

- **Owns:** [*the interface register between software and embedded — every agreed dependency, in one documented place*].
- **Why this role exists:** under the current structure, interface changes are communicated ad hoc between two leads, which the interviews link directly to [*late integration failures — cite App. 1.5*]. A single owner means every interface change has one documented point of record.
- **Who fills it:** [*not a new hire — reallocated from an existing member's time; state whose and how many hours*].

Software Integration Manager

- **Owns:** [*the interface register between software and embedded — every agreed dependency, in one documented place*].
- **Why this role exists:** under the current structure, interface changes are communicated ad hoc between two leads, which the interviews link directly to [*late integration failures — cite App. 1.5*]. A single owner means every interface change has one documented point of record.
- **Who fills it:** [*not a new hire — reallocated from an existing member's time; state whose and how many hours*].

Technical Integration Manager

- **Owns:** [*the interface register between software and embedded — every agreed dependency, in one documented place*].
- **Why this role exists:** under the current structure, interface changes are communicated ad hoc between two leads, which the interviews link directly to [*late integration failures — cite App. 1.5*]. A single owner means every interface change has one documented point of record.
- **Who fills it:** [*not a new hire — reallocated from an existing member's time; state whose and how many hours*].