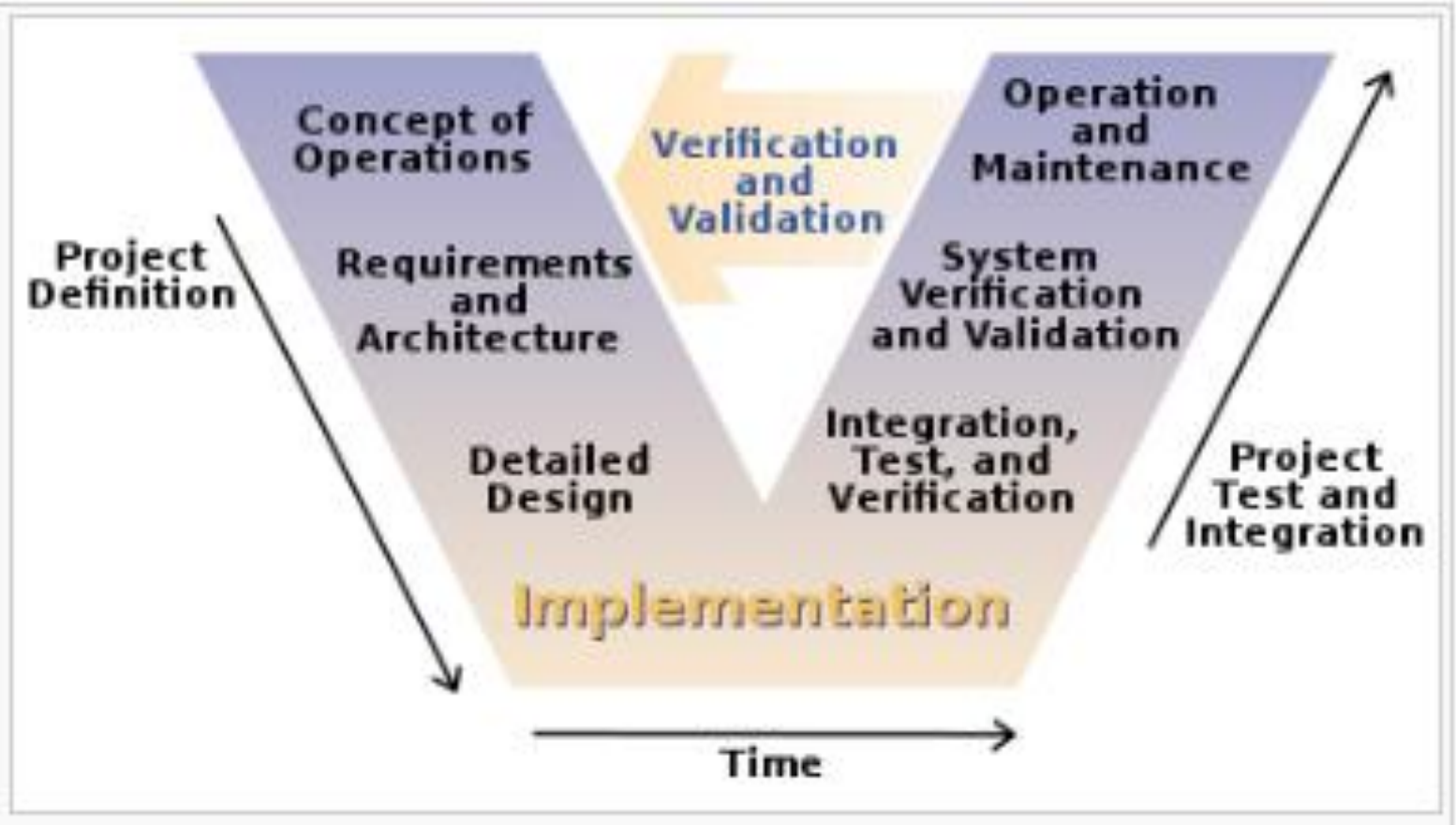
Systems Engineering Training

A **system** is a set of interacting or interdependent component parts forming a complex/intricate whole.

Systems Engineering V Diagram



Definitions:
Architecture - the complex or carefully designed structure of something
Integration - to put together parts or elements and combine them into a whole
Verification - the act or process of confirming or checking accuracy
Validation - to make something officially acceptable or approved

Systems Engineering Training: Requirements

What are the characteristics of a good requirement?

- Verifiable
- Clear and concise
- Complete
- Viable
- Necessary
- Implementation free

Systems Engineering Training

A verifiable requirement ...

- is stated in such a way that it **can be tested** by:
  - inspection,
  - analysis, or
  - demonstration.

A clear & concise requirement ...

- must consist of a **single requirement**,
- should be no more than **30-50 words** in length,
- must be **easily read and understood** by non technical people,
- must be **unambiguous** and not susceptible to multiple interpretations,
- must **not contain** definitions, descriptions of its use, or reasons for its need, and
- must **avoid** subjective or open-ended terms.

A complete requirement ...

- contains **all the information** that is needed to define the system function,
- leaves **no one guessing** (For how long?, 50 % of what?), and
- includes **measurement units** (inches or centimeters?).

- **Bad example:**
  - UR1: The system must be user friendly.
  - **How should we measure user friendliness?**

- **Good example:**
  - UR1: The user interface shall be menu driven. It shall provide dialog boxes, help screens, radio buttons, dropdown list boxes, and spin buttons for user inputs.

- **Bad example:**
  - UR2: All screens must appear on the monitor quickly.
  - **How long is quickly?**

- **Good example:**
  - UR2: When the user accesses any screen, it must appear on the monitor within 2 seconds.

- **Bad example:**
  - UR3: On loss of power, the battery backup must support normal operations.
  - **For how long?**

- **Good example:**
  - UR3: On loss of power, the battery backup must support normal operations for 20 minutes.

Systems Engineering Training

A viable requirement ...

- can be **met using existing technology**,
- can be **achieved within the budget**,
- can be **met within the schedule**,
- is something the organization has the **necessary skills to utilize**,
- will be **used by the end users**, and
- must be **helpful to build the system**.

A necessary requirement ...

- is one that **must be present to meet system objectives**, and
- is **absolutely critical** for the operation of the system,
- leads to a **deficiency in the system if it is removed**.

A requirement that is free of implementation details ...

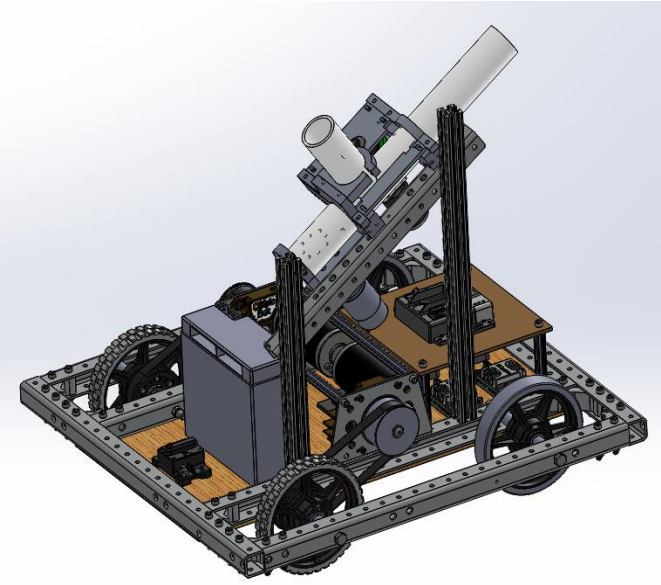
- defines **what functions are provided** by the system,
- **does NOT specify how** a function can or should be implemented, and
- allows the **system developer to decide what technology is best** suited to achieve the function.

• **Bad example:**	• **Good example:**
- The replacement control system shall be installed with no disruption to production. - **This is an unrealistic expectation.**	- The replacement control system shall be installed causing no more than 2 days of production disruption.

• **Bad example:**	• **Good example:**
- All desktop PCs for the project must be configured with 512MB of memory, DVD ROM/CD-RW multifunction drive and a 21-inch flat screen monitor. - **This may not be needed for all PCs for the project.**	- The desktop PCs for the developers on the project must be configured with 512MB of memory, DVD ROM/ CD-RW multifunction drive and a 21-inch flat screen monitor.

• **Bad example:**	• **Good example:**
- After 3 unsuccessful attempts to log on, a Java Script routine must run and lock the user out of the system. - **Specifying a JavaScript routine concerns how the requirement will be implemented.**	- After 3 unsuccessful attempts to log on, the user must be locked out of the system.

Yr1 Robot: Features and Functions



Definitions:
Features - a distinctive attribute or aspect of something
Functions - an activity or purpose natural to or intended for a person or thing

What are the key features and functions of the Yr 1 Robot?

- Controllable by wireless joystick controller
- Drives forward, backwards, left, or right
- Follows a white line autonomously
- Shoots ping pong balls at a target
- . . . .

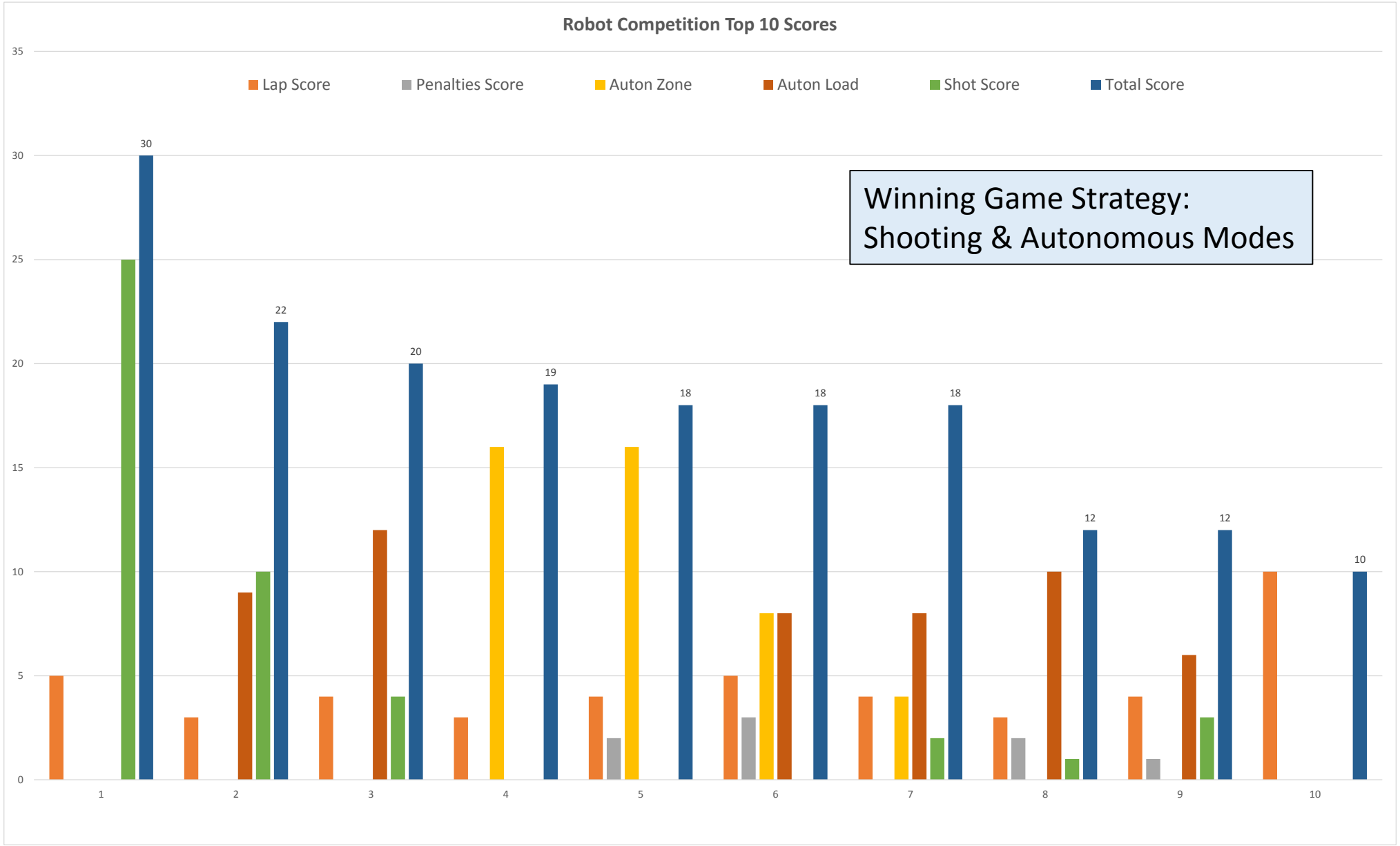
What are the key features and functions of the Yr 1 Robot **Shooter**?

- Shoots ping pong balls at a target
- Holds 6 or more ping pong balls
- . . . .

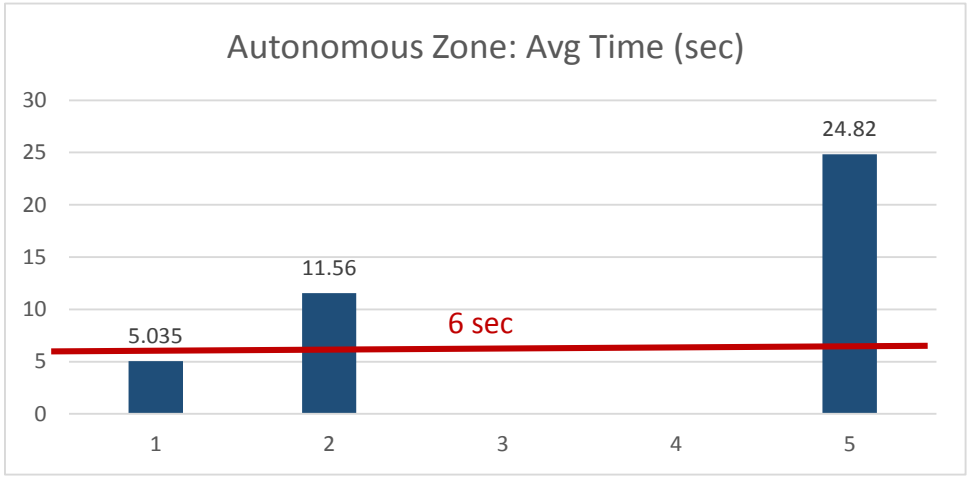
What are the key features and functions of the Yr 1 Robot **Line Tracker**?

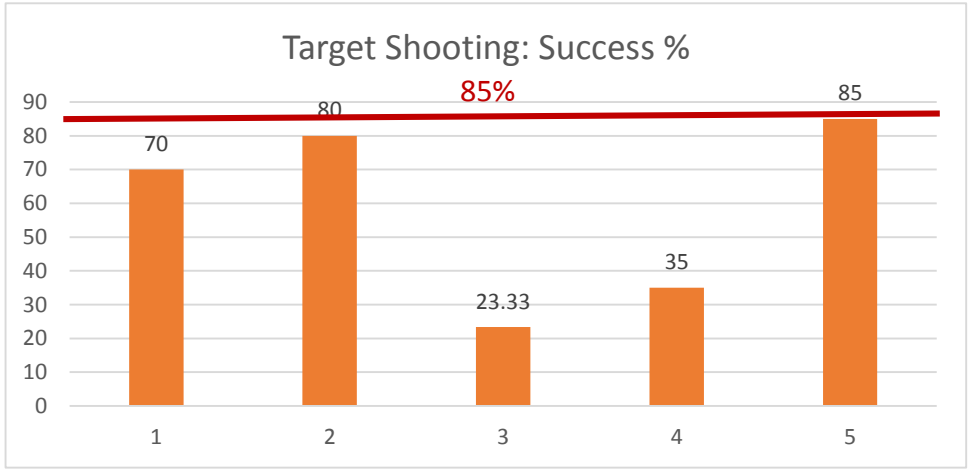
- Follows a white line on the floor
- Robust attachment to bottom of the robot
- . . . .

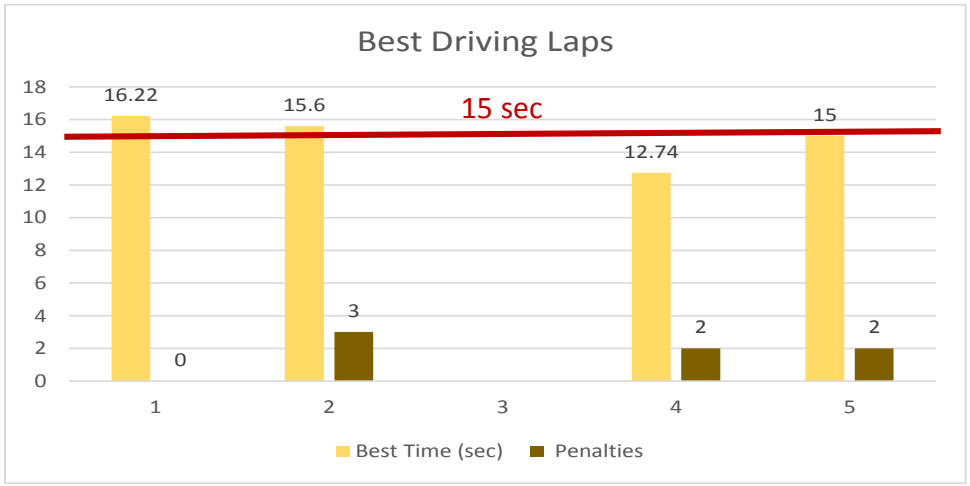
Yr1 Robot: Competition Results

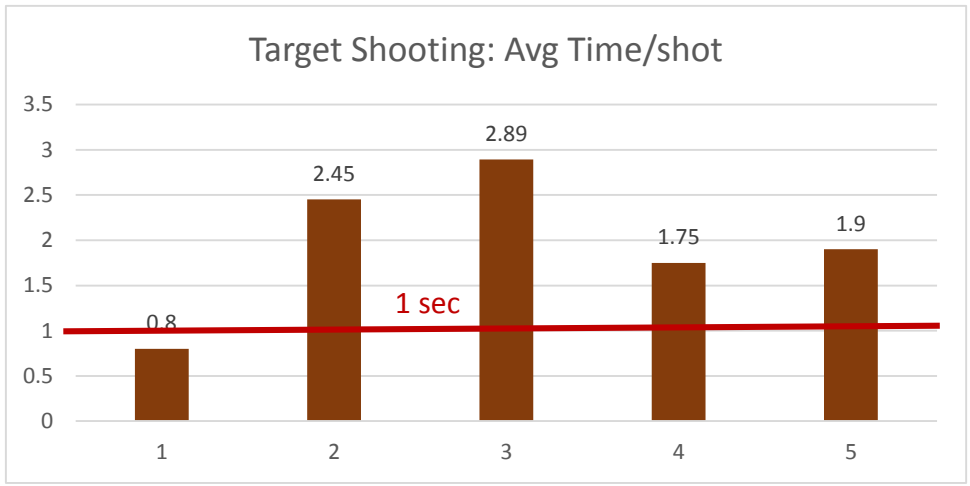


Yr1 Robot: System Test Results



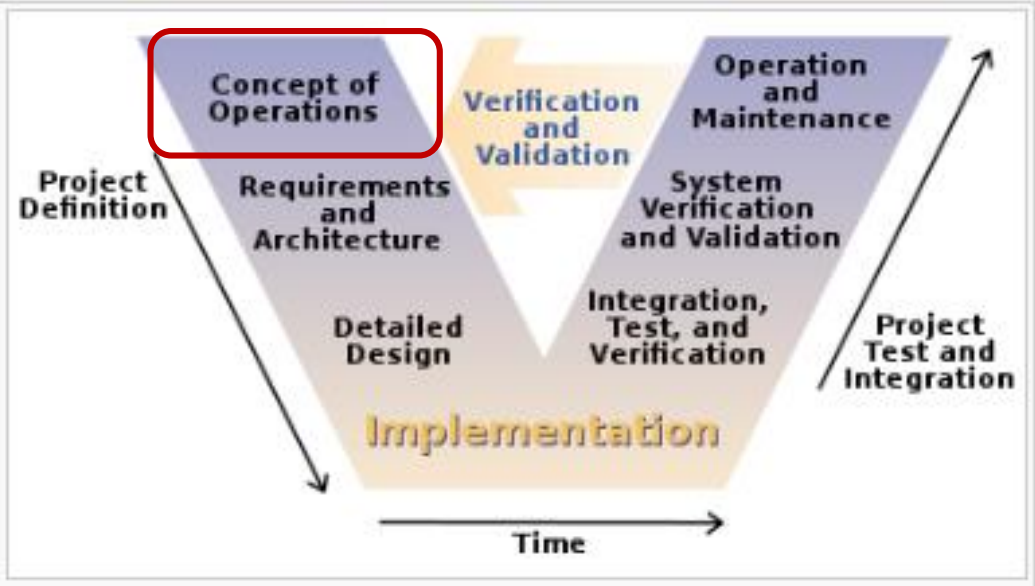


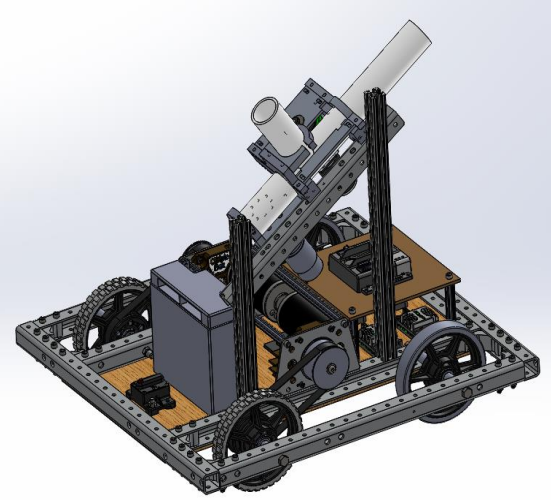




Targets selected based on benchmarking of 2015 Yr1 Robots (system tests).

Yr1 Robot Concept of Operations



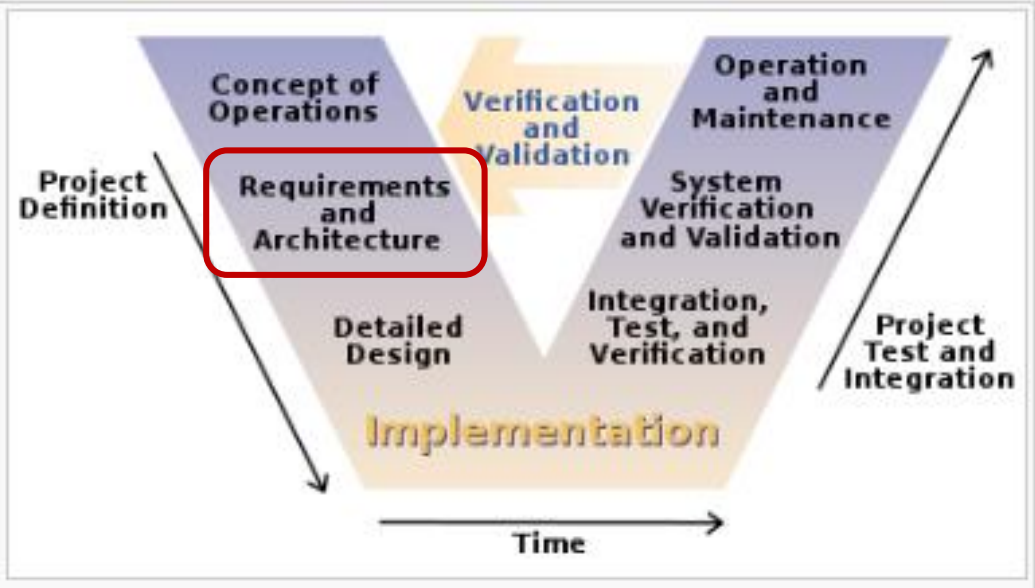


"High Level Requirements" or "Concept of Operations" define WHAT is needed (not HOW to do it).

Concept of Operations:

- The robot shall adhere to all of the Ping Pong Pandemonium Rules
- The robot shall be capable of driving the game field in teleop mode (wireless joystick control)
- The robot should be able to drive one loop of the field in under 15 sec.
- The robot should be capable to hold up to six ping pong balls at a time
- The robot should be capable to shoot at a rate of 1 shot per second
- The robot should be capable of hitting the game target with an accuracy greater than 85%
- The robot should be capable to navigate the autonomous zone in under 6 seconds

Yr1 Robot Architecture

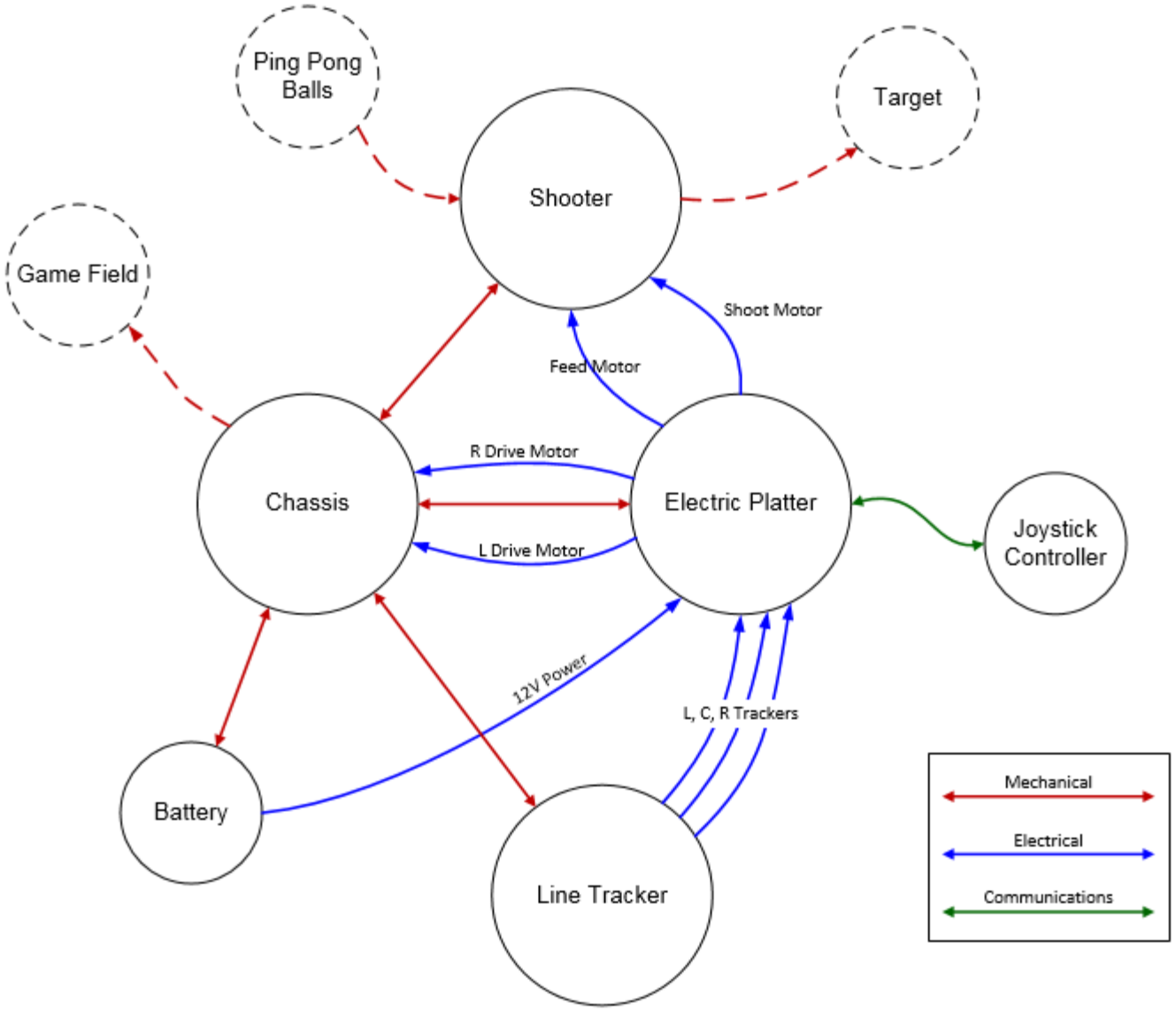


From the Concept of Operations, determine the robot architecture and subsystem requirements.
These requirements include both 'WHAT is needed' and some 'HOW to do it'.

Breakdown the system into 4 main subsystems . . .

- Chassis
- Electric Platter
- Shooter
- Line Tracker

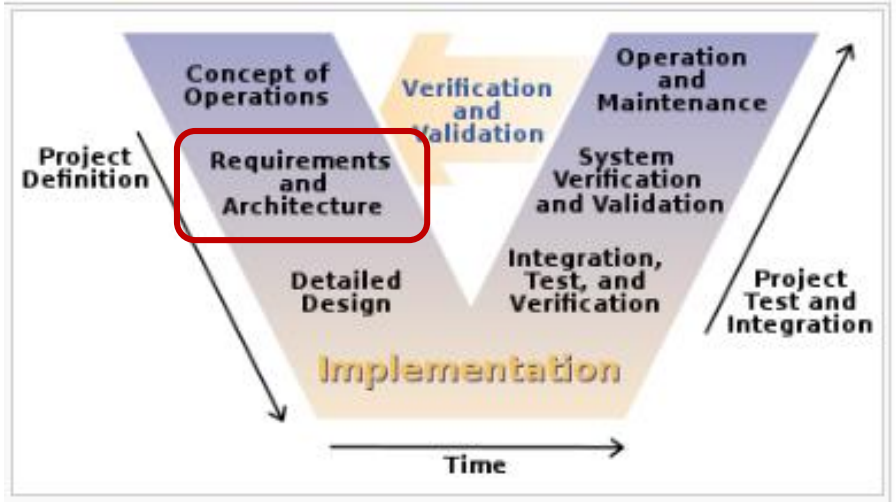
Context Diagram: Yr1 Robot



The context diagram is a very effective means to illustrate components of a system (or subsystem) and interactions between components/subsystems. The context diagram can help identify functions, interfaces, and requirements.

From the context diagram, detailed design can begin. Based on this context diagram, how many motors are required?

Subsystem Requirements

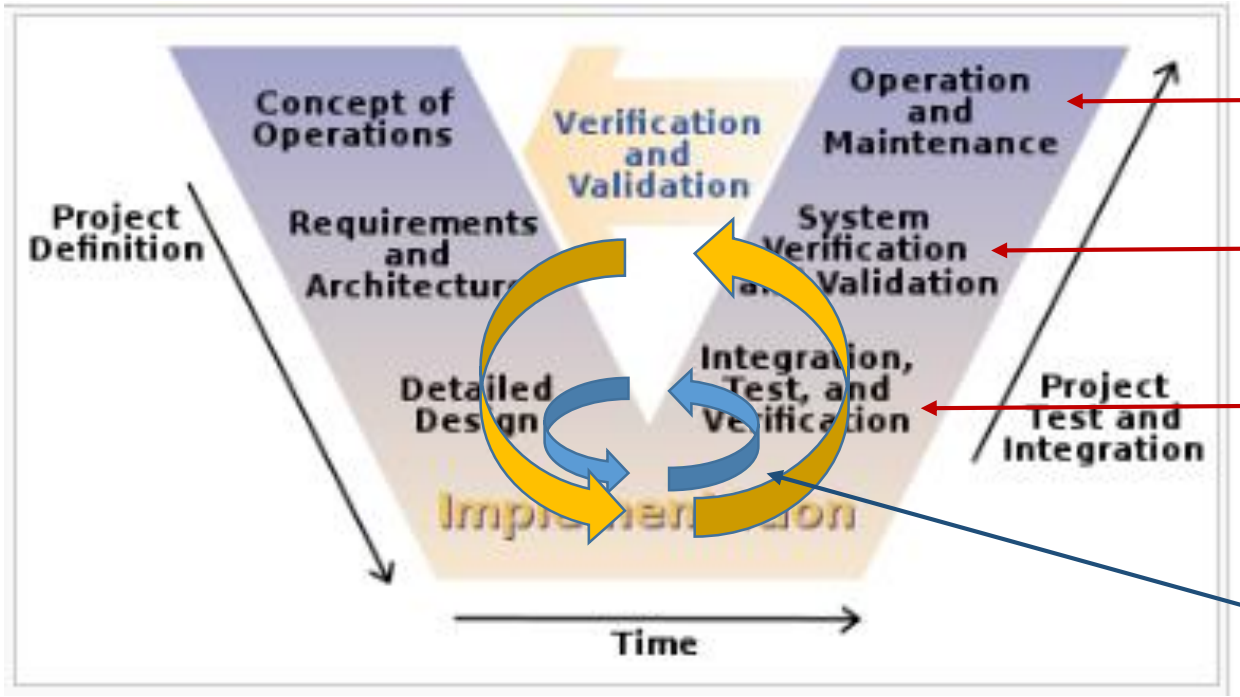


- Chassis - What is needed:
  - ❖ How: Frame
  - ❖ How: Powertrain selection
  - ❖ How: Wheel size & type
- Shooter - What is needed:
  - ❖ How: Mounting options
  - ❖ How: Modifications (optimization) for accuracy improvements
  - ❖ How: Ball storage
- Line Tracker - What is needed:
  - ❖ How: Mounting options
  - ❖ How: Autonomous Software

Constraints:

- Budget: limited new purchases to be determined by lead mentors (new controller has been purchased)
- Materials: primarily what is available in Team 302 inventory
- Time: reference project plan

Verification and Validation



Example: does the robot drive autonomously in the required timeframe?

Example: does the shooter meet or exceed the accuracy requirement?

Example: does the shooter feed motor load successive balls in the required timeframe?

Iterate design and test at component level until requirements are met.

Even before the Detailed Design work is completed, component/subsystem and system verification tests can be defined. Wherever possible, component and subsystem level tests should be performed before system level testing. Iterate at the design and test at the component level until requirements are verified before moving up the “V”.