Performance Measurement and Software QA in the RoboCup Standard Platform League: A qualitative interview study

Thomas Klute

Robotics Research Institute, Section Information Technology,
TU Dortmund University, 44227 Dortmund, Germany
thomas.klute@tu-dortmund.de
https://www.irf.tu-dortmund.de/

Abstract. The purpose of this paper is to examine how teams in the RoboCup Standard Platform League (SPL) approach software quality assurance and performance measurement. RoboCup serves as an experimental platform for students to delve into robotics research. However, with the ambitious goal set by RoboCup, teams must continuously enhance their skills and competitiveness. Our study adopts a qualitative descriptive approach, involving interviews with team leaders and designated team members from seven different RoboCup Standard Platform League teams. During these interviews, we recorded field notes and subsequently conducted qualitative content analysis on the gathered data. The findings indicate that nearly all teams rely on expert judgment during test games conducted in their laboratory settings for quality assurance and performance assessment, including the most successful teams in the league. However, only a minority of teams have implemented more structured and automated mechanisms akin to those used in software engineering. Furthermore, almost none of the teams engage in performance measurement beyond manual assessment through observation. We explore potential explanations for these findings and conclude that there is ample room for improvement towards implementing more structured and automated approaches to quality assurance and performance measurement, both at the team level and within the league itself. These results hold practical significance for shaping the roadmap of RoboCup and offer valuable insights for establishing team-based quality assurance and performance measurement practices.

Keywords: RoboCup SPL · Performance measurement · QA · Software testing

1 Introduction

RoboCup as an organization set an ambitious goal 25 years ago: to beat a human soccer team by 2050 [1]. Now, halfway to that goal, there has been tremendous progress made, starting with simulated and wheeled robots [2], then using the

Sony Aibo dogs [3] up to a point where we are now with humanoid robots being able to demonstrate basic skills successfully, including some first elements of team play. There are still 25 years left for research and development work, moving towards more advanced skills, including team play, and achieving the goal of being able to play (and potentially win) against a team of human football players. Meeting this goal depends on continuous progress and improvement in the RoboCup leagues and mainly depends on the research and development among the teams.

On the other hand, RoboCup has been serving a field for student education in robotics from the beginning [4]. New students enter this complex field each year, joining one of the RoboCup teams and aiming to engineer high-quality robotic systems. Balancing the need for student education with the demands of building top-notch robots poses challenges. RoboCup teams must maintain the quality and performance of their already achieved skills at an appropriate level while focusing on more recent research areas that include more sophisticated team behavior and tactical components of the game.

Given these competing goals, it is crucial to understand how RoboCup teams navigate this landscape and what strategies they employ to maintain and enhance their playing strength.

2 Background

RoboCup serves as a research hub for robotics, aiming to advance robotic skills for the broader research community. Additionally, it plays a significant role in student education, inspiring young people to engage with robotics and push the boundaries of scientific knowledge. Most RoboCup teams comprise both researchers and students, all striving for top scores in their respective leagues. However, robot hardware, along with the control software required to operate robots safely and effectively, poses significant challenges. Numerous modules must function seamlessly within defined parameters to enable the complex behaviors necessary to fulfill tasks, all while ensuring the safety of both robots and potential human players. Safety requirements have already been specified for industrial robots in ISO 10218 [5] and cooperative industrial robots in ISO 15066 [6]. However, for soccer playing robots in an environment shared with human soccer players currently no specification exists.

While tasks like humanoid robot locomotion, image processing, and position tracking are challenging yet solvable, they will soon become fundamental skills for every RoboCup team. As we approach 2050, the emphasis will shift towards faster, more agile movements, sophisticated ball handling, better passing and more vigorous kicking, audiovisual communication, and coordinated teamwork among multiple robots - all within strict safety constraints to ensure the well-being of all participants.

We notice that existing teams and newcomers struggle to keep pace with the evolving complexity of the league, prompting questions about better strategies

to help them keep up with the other teams and maintaining or enhancing the quality of their software over time.

Given that much of RoboCup revolves around student education, with many participants starting from scratch in robotics knowledge, it is essential to understand how teams can maintain and improve their competitiveness in these areas over time.

3 Methods

3.1 Aim

The study aims to collect examples of how RoboCup Teams approach their software development tasks in general and what methods they use for quality assurance and performance measurement.

3.2 Research Questions

Our primary research questions are:

- What code base and tools do the teams use in the SPL, and how successful are they?
- What approaches do RoboCup teams in the SPL use to maintain or improve their playing strength?
- Do SPL RoboCup teams use a structured software development life cycle (SDLC) and quality assurance (QA) cycle approach?
- Do SPL RoboCup teams perform continuous performance/quality metrics measurement and monitoring?

3.3 Design

To gather accurate responses to our research questions, we initially asked the teams, who among them is responsible for quality control and performance benchmarking. If this role was not explicitly filled out by a team member, we asked for a team leader who at least possessed a thorough understanding of how these matters were addressed within the team.

We structured the interview into sections, with the first section asking for the personal and scientific background of the participant, their role in the RoboCup team, and research work done in the context of the team.

The second section of the interview focused on the RoboCup team, tournament placements, the code base, and the team's research activities. We also asked what kind of simulator software the team uses and how accurately they consider the simulator to perform compared to reality.

The third section of the interview asks about the team's approach to quality assurance and performance measurement, test coverage, and CI pipelines used. We also asked whether they adhere to a structured software development cycle.

The fourth section of the interview focuses on software testing and performance measurements in relation to the RoboCup team and asks for the proposed next best steps for the team and the league.

As a last section, we presented a potential approach to automated software testing and performance measurement, considering the specific requirements of RoboCup teams, and asked for comments.

3.4 Data collection

Data was collected through interviews of 60 to 90 minutes length at the German Open Replacement Event (GORE) in Hamburg and during the RoboCup 2023 in Bordeaux. The interviewer took field notes during the interview. Other data not answered or unknown by the participants, such as the number of scientific publications over the years or the team's rankings in the last competitions, were added or validated using other sources, such as the official RoboCup website and the teams' websites. That raw data was manually processed later to extract and summarize the relevant facts and statements.

We selected teams based on their availability during the competition and their willingness to take part in the survey. Second, we wanted to include teams that might have certain mechanisms in place that we want to identify in this study. Thus, we included some teams that demonstrated appropriate playing strength over the last years. The selection included the two best teams over the last years (B-Human and HTWK) and three teams often ranked in the top eight of competitions (Bembelbots, HULKs, and our team, the Nao Devils) over the last years. The availability of the relevant team members during the competition days limited the number of teams and interviews.

4 Data analysis and results

4.1 Participants, teams and education

We conducted seven interviews with relevant people from seven RoboCup teams (B-Human, Bembelbots, Dutch Nao, HTWK, HULKs, Naova, and our team, the Nao Devils), with all teams participating in the RoboCup Standard Platform League (SPL). All teams are part of educational institutions on a university level (University of Bremen/DFKI, University of Frankfurt, University of Eindhoven (NL), HTWK Leipzig, University of Hamburg, TU Dortmund, and ETS Montreal (CA)). The roles of the participants were Team-Leads (3), Tech-Lead (1), Dev-Lead (1), and Org-Lead+Developer (1). Their current academic rank ranged from Student (1) over Bachelor (3) and Master (1) to Phd (1), all from the field of Computer Science (4), Artificial Intelligence (1), and Software Engineering (1). Most did not write any thesis on RoboCup (6) except for one master thesis. Some have written or are writing papers on RoboCup topics (3). The time spent being part of the RoboCup community ranged from 2-5 years (3), 6-10 years (1), 11-15 years (1) to more than 20 years (1).

Table 1. RoboCup placements of surveyed teams.

	Code base		Placement[1]				
Team	origin	lang	2017	2018	2019	2022	2023
B-Human	1998	C++	1	2	1	1	1
HTWK	2009	C++	2	1	2	2	2
HULKs	2021	Rust	5-8	4	5-8	5	4
Nao Devils	2015[2]	C++	3	5-8	4	4	5
Bembelbots	2009	C++	13-16	9-12	5-8	7	8
Dutch Nao	2023	Rust	9-24	9-20	9	10	C4[3]
Naova	2017[2]	C++	-	13	17	13	C6
# of teams			24	21	20	13	17

4.2 Code Base

The code base of the teams interviewed is mainly written in C++ (5), followed by Rust (2). Rust has recently evolved into a popular programming language with different advantages over C++; it has become an alternative when rewriting a code base from scratch or migrating existing code. This finding correlates with the age of the code bases of the teams: The two teams having 1-5-year-old code bases are written in Rust; the others (6-10 years (2), 11-20 years (2), and ¿20 years (1)) are all written in C++. Four teams wrote their code base from scratch; the other three teams used a fork or partially forked existing code. Remarkable is that parts of the most successful code base of B-Human, which is often used as the basis for forks by newcomer teams, are over 25 years old and are based on software written to simulate electric wheelchairs [8]. The sizes of the code bases are also quite different and range from 5k lines of code (1), <50k (2), 150k (1) up to 240k (1).

4.3 Simulators

The 3D simulation environments most often used are WeBots (3) and SimRobot (3), the simulator contained in the B-Human code base. Other simulators used are LoLa (1) and Copelia (1). Some teams (3) have implemented a separate 2D simulator that reduces the motion part of the simulation to simple 2D movements. Thus, it is often used to simulate tactical aspects of the game. When asked about the accuracy of their simulation environments compared with reality, some teams (3) say that their simulator is not accurate, mainly regarding the physics simulation and the applied motion control. Two teams said the environment is quite good but not perfect (2), one named it quite accurate, and one team could not judge on that question. When asked if teams spend efforts to close the gap between reality and the simulator, all teams denied this. One team

[1] Years 2020 and 2021 omitted. No RoboCup was held due to COVID-19.
[2] based on a fork of the B-Human code base.
[3] Cx means a Challenge Shield placement, the 2nd SPL league.

works on a 2D simulator to better simulate the behavior. We asked the teams if their simulator environments are capable of running scripted tests that measure correct behavior and certain defined KPIs. The answers were: partially, but not automated (1), no (4), or only in the 2D simulators and only for the values that can be measured in the 2D simulator.

4.4 QA measures

When asking for QA measures in place to track and improve or at least keep the quality and playing strength of a team over the years, the teams came up with various answers: The most often mentioned measures are test games, including expert judgment (5) and code reviews based on merge requests (5). Playing test games in the lab environment or during competitions and judging the results and events identified during such test games seems to be a widespread practice among the teams. Code reviews are also a well-known QA measure in software engineering. The teams use feature branches of their version control to develop new features or extensive rework efforts and then create merge requests that become subject to code reviews before being merged into the master branch. A CI pipeline and automated software tests was mentioned by two teams, although one of them only uses a basic smoke test as the only software test, where a simulated test game is started, and a goal has to be scored within a certain amount of time. Three teams mentioned using at least parts of the Scrum methodology, including regular and standup meetings. Other teams mentioned that they tried Scrum, but it turned out to be unsuitable, or they considered it of little help. Two teams mentioned ticket systems and boards to organize and prioritize their tickets. Two teams explicitly mentioned that they use the simulator for testing developments. Only some teams cover parts of their code using software tests. Four teams have no or less than one percent test coverage in place. One team mentioned having below 10 percent of code coverage and not running these tests automatically. Two teams seem to use software tests more extensively, with parts of their code fully covered and others partially covered.

4.5 Benchmarking and performance measurement

We asked the team if they benchmark their software and if they measure certain performance indicators to judge on the performance of specific skills, like for example walking speed or the ability to kick a ball into the goal. Three teams answered that they do not have such measures in place, three others mentioned that they only do some manual comparison as part of the expert judgement. One team answered, that they do some measurements on walking, but these are very basic. Mainly they are doing manual testing on the field. One team draws some statistics from the game controller logs of tournament games.

4.6 Roadmap + KPI for SPL

We mentioned that the Humanoid League of RoboCup has proposed a Roadmap for the league to measure specific performance indicator values during the com-

petition games. These performance indicators would be used to detect progress of the league's teams and trigger rule changes once specific KPI values are reached. Such a mechanism can be used to foster teams' progress and adapt the league's framework conditions according to the team's progress. We also mentioned this initiative and asked the participants if this would be a feasible approach for the SPL. We received positive results, such as: 'Yes, progress should actually be made measurable' and 'Has been discussed in the tech committee for three years. The very good teams want the league to progress.'. However, problems with this approach were also mentioned: 'The league has lost 1/3 of its teams since COVID. Apart from teams from Germany, there are too few teams being founded. This is an argument against making the league harder.'

4.7 Next best steps for the teams

We asked the participants for the next best steps for their teams. Two teams stated that more resources and more developers are an urgent need. Improving different aspects of the vision system, like line recognition and sensor fusion, was mentioned three times. Improving robot skills, such as better passing and a faster goalie, was mentioned once. Removing the need for robot calibration to adapt the robots to different lighting and playground conditions was also mentioned.

All in all, the answers show that improving the robot software to better adapt to different playgrounds and lighting conditions and expanding the team to include more developers are the most important topics.

4.8 Next best steps for the league

When asked about the next best steps for the SPL league, the participants mentioned several interesting ideas: The idea of a shared simulator was mentioned two times, where teams could virtually play against each other without dealing with technical issues and robot hardware wear-off. One participant voted against an additional shared simulator because, from his point of view many teams are already limited by time and human resources, and thus an additional task of adapting and maintaining integration of their code bases to an additional simulator could lead to overstrain.

Two participants mentioned that a general roadmap for the league would be beneficial. Two participants mentioned that the goal was to start measuring performance and measurability in general to identify the league's progress. Additionally, the game states 'initial' and 'ready' could be removed as robots become more autonomous and can enter the pitch automatically, identify whistle and referee gestures to lead the game. The game controller software currently guiding the game could be removed later.

One participant mentioned improving team play. Team play between robots requires coordinated behavior, and passing the ball between robots has already been introduced as one challenge for the league. Currently, '1 or 2 teams at maximum are capable of playing controlled passes' and this should be extended. The

amazement of external spectators was also mentioned as a reason why improving team play would be beneficial. One participant mentioned that publishing the complete source code, including additional tools, could be beneficial. The teams participating in RoboCup must partially publish their source code after participation [7, p. 43]. However, parts of the source code are permitted to be omitted.

One idea was to migrate to different and better robots as soon as better hardware at a reasonable price becomes available, to overcome the current limits of the league set by the robot hardware.

5 Findings

First, we need to note that the two top teams of the SPL of the last years (B-Human and HTWK) have been able to keep up their playing strength and defend their placements, given the first and second places they have achieved constantly during the last five years (volatility of 0.4). This top group is followed by a group of three teams (rUNSWift, HULKs, and Nao Devils) that very often achieve third or fourth place or at least are amoung the teams on the 5th-8th rank (13 of 15 placements on 3rd, 4th and 5-8th rank over 5 years). However, the volatility of their placements is about two to three times higher. rUNSWift has consistently achieved third place in the last three RoboCups and may thus be considered one of the top teams from 2019 on.

It seems that - based on these final placements - the best teams are able to keep up their playing strength or at least reach appropriate playing strength in the relevant games of the RoboCup to defend their placement. Given the consistent performance of these top teams, it's crucial to delve into their approaches to quality assurance (QA). This aspect of their strategy could be a key factor in their sustained success.

5.1 Approaches to maintaining and improve playing strength

The teams use a variety of approaches. The most often used methods are expert judgment of test games and code reviews based on merge requests.

Expert judgment based on watching a test game can be seen as an end-to-end test in which test results are not automatically measured but manually estimated and assessed. This seems to work well for the majority of teams. The popularity seems to be because such a test does not need any additional software development or infrastructure besides what is needed for tournaments - a pitch and a set of robots.

Merge requests and code reviews are common practices in software engineering and are often used in software development lifecycles. Thus, we can conclude that the teams actively use certain aspects of software engineering practice. However, the lack of software tests, in general, shows that this is not a high-priority topic, and teams seem to accept the downside of not having software test coverage.

5.2 Structured software development lifecycle

Most teams have adapted some elements of software development methodologies, such as SCRUM. These elements, like daily standup meetings and weekly planning, are mainly used for team communication and coordination. Besides that, the teams do not follow structured processes of software development lifecycles, as, for example, often used in industrial software development[9].

5.3 Performance and quality metrics measurement

We can summarize that some measurements and statistics seem to be used. However, these are, in all cases, not automated but require some manual processing. So, in summary, one can say that there is no comprehensive measurement of performance indicators in place, and additionally, such indicators are not subject to any monitoring that would indicate improvement or degradation.

6 Discussion

6.1 State of the SPL league

One remarkable result of the interviews and our observations is that the best teams of the SPL league have surpassed the fundamental problems of robot soccer and can demonstrate advanced skills like passing the ball between human-robot team members in a controlled and repeatable way, using this as part of their game tactic. They can keep their playing strength by using ad hoc expert judgments to detect issues and monitor their team's progress. These teams seem to be ready to proceed further by adding more sophisticated game tactics and advanced skills to their gameplay while keeping their basic skills at an appropriate performance level. However, the number of teams demonstrating such skills is still low.

Other teams are still struggling with basic issues like line detection, self-location, and walking/kicking motion control. It becomes evident that the majority of teams are still struggling with more or less basic skills, as many of the next best steps for the team are still covering basic skills and additionally taking into account that the majority of teams we interviewed typically reach placements within the top 5 of the SPL league,

Based on our experiences, the playing strength in the final game does not necessarily mean that the playing strength has been on the same level during all of the teams' games during the tournament. The playing strength of the final games may result from the optimization work of the team during the tournament.

The yearly code release is a good chance to keep up with the top teams by integrating their source code into problematic modules or replacing them. However, from our experience, the complexity of porting sources from one code base to another is relatively high. Thus, teams might not consider migrating code as their preferred solution when addressing problematic modules.

6.2 Approaches to QA and performance measurement

Robot soccer is a game that requires basic modules and skills to function correctly in order to reach a level of gameplay that deserves the title. Playing and judging test games is the most natural way to prepare for a soccer tournament. Even kids can make a basic judgment of how well the robots perform by watching a test game. An interesting finding is that the two very successful teams also rely mainly on expert judgment. Thus, we must conclude that this strategy is successful if appropriately applied.

However, there is much room for improvement: Expert judgment is subjective and not normalized. Without normalization, comparing several assessments in detail over time is difficult. Automated measurements of key performance indicators over time would enable teams to monitor their performance and detect improvements or degradation. Additionally, if different teams use the same performance indicators, they could compare their skills with what other teams can achieve. In summary, we can conclude that the RoboCup SPL league teams have not focused on this field, which leaves much room for improvement in the future.

Implementing a benchmarking of appropriate metrics measuring aspects of playing strength is judged as very valuable for the teams and for the whole league by almost all participants.

Thus, we propose that the SPL league defines basic metrics of playing strength suitable for measuring the overall league's progress. The league should provide tools to measure such metrics. These tools should be independent of any specific implementation or framework so that all teams can set up and use them without much effort. The tools should offer an easy way to add additional team-individual metrics. This would offer the teams an easy way to set up an automated approach to quality control and enable benchmarking to track their individual progress.

6.3 Limitations

In this study, we interviewed only SPL teams. Other RoboCup leagues and their approaches were not researched, although they might have been interesting. However, in this study, we only wanted to explore the landscape of SPL teams and their approaches. Additionally, we did not interview every SPL team; we only interviewed a selection of teams of interest to our research and were available when we did the interviews. Therefore, this study and its results must be interpreted with the restriction that they involved only a subset of teams. The study's overall aim was to gain insights into the teams' approaches, and this was achieved.

6.4 Conclusions

The survey shows that the teams currently use simple and mostly manual strategies to assess their quality, performance and playing strength. Structured software quality and performance measurement is uncommon and has yet to be

introduced by almost all teams. Most teams perform expert judgment by observing the robots during test matches, a simple but effective approach to quality assurance. One reason for not introducing more automated and structured QA approaches seems to be the initial effort to set them up and update them as development progresses. Another reason seems to be the lack of resources to tackle such issues, combined with a low prioritization as other features - relevant to the game - are the focus of the teams. We suggest that RoboCup, in this case, the SPL league, define basic metrics of playing strength and provide teams with tools and methods to measure and monitor such performance indicators.

References

1. RoboCup website, archived version of December 1998 https://web.archive.org/web/19981201174356/www.robocup.org/overview/22.html.
2. RoboCup History, https://www.robocup.org/a_brief_history_of_robocup. Last accessed 10. Feb 2024
3. RoboCup SPL History, https://spl.robocup.org/history/. Last accessed 10. Feb 2024
4. Verner, I.M.: The Survey of RoboCup' 98: Who, How and Why. RoboCup 1998. Lecture Notes in Computer Science, pp. 109-119 (1999) https://doi.org/10.1007/3-540-48422-1_8
5. ISO 10218-1:2011: Robots and robotic devices - Safety requirements for industrial robots. International Organization for Standardization, Geneva, Switzerland. https://www.iso.org/standard/51330.html
6. ISO/TS 15066:2016: Robots and robotic devices - Collaborative robots. International Organization for Standardization, Geneva, Switzerland. https://www.iso.org/standard/62996.html
7. RoboCup SPL Rules, https://spl.robocup.org/wp-content/uploads/SPL-Rules-master.pdf. Last accessed 10. Feb 2024
8. Laue, T., Röfer, T.: SimRobot - Development and Applications. In: Workshop Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2008). http://www.informatik.uni-bremen.de/kogrob/papers/SIMPAR-Laue-Roefer-08.pdf
9. Gupta, A.: Comparative Study of Different SDLC Models, In: iJRASET International Journal for Research in Applied Science and Engineering Technology (2021) https://doi.org/10.22214/ijraset.2021.38736