Chapter

Robots, Everlasting? A Framework for Classifying CS Educational Robots

Ishin Iwasaki, Corben Roszak, Parama Chaudhuri, Katherine LaRue and Caroline D. Hardin

Abstract

Educational robots are an exciting and growing field. While some (Lego Mindstorms, for example) have been around for decades, most are only a few years old and their durability is untested; exacerbating this are those only usable with apps, that may become suddenly unavailable. This has created a nascent but significant problem: schools investing significant time and money for educational robots with little ability to know if they will work for years or just days. Other fields in science, technology, education, and math (STEM) beyond computer science also encounter this issue as more educational robots and apps for those disciplines permeate the market. While this chapter analyzes this issue from a CS perspective, the lessons learned can be applied to other STEM areas. This chapter explores the history of the problem, documents several examples of devices that have succumbed, details the unique and specific needs of school customers, and introduces the Computer Science Risk Analysis Framework for Toys (CS RAFT) to help teachers and schools evaluate a device purchase based on a holistic understanding of device longevity. This study will also provide recommendations for CS and STEM educational robot designers.

Keywords: robotics, sustainability, tangibles, education

1. Introduction

Computer science plays a critical role in our technologically connected world, and students will get exposed to many technologies within their early years of schooling [1]. Computational thinking (CT) has been used to introduce students to computing principles within the context of the subject areas they are learning, which helps make it relevant to their understanding [2]. Robots for computer science education is a rapidly blossoming field, growing from a handful of devices pioneered by people like Michael J. Freeman in 1974 and Seymour Papert in the 1980s to over 80 different modern devices available today, many with expansive ecosystems of teacher and user resources [3, 4]. Teaching abstract concepts of CT through robotics is especially helpful as it makes the abstract concrete and children can observe the direct results of their programming commands on the robot's actions [5]. What distinguishes these robots from robots designed for tasks like manufacturing is that they are specifically designed and marketed to help teach introductory computer science concepts to K-12 students. They are typically small, cute, brightly colored, and have heavily anthropomorphic features. While there are some overlaps with robots designed for hobbyists, CS and other STEM Ed robots are oriented not around accomplishing a task for the task's sake, but in learning concepts through built-in puzzles or challenges. CS Ed robots are designed for children who might have no computer science experience and are helped by adults with limited CS experience. In a decade where significant efforts have been made to battle the stereotype of computer science and other STEM fields as intimidating, these educational robots are friendly, fun, and a great entry point for beginners [5].

CS Ed toys and robots are not only marketed for home use; they are also marketed to teachers and schools where they will reach more children and have a greater impact. Following a decade of breathless Kickstarter campaigns for CS Ed robots, a nascent problem is that some of the robots which are sold as preparing children for the future will not see the future themselves. These robots with volatile futures are likely to become unusable within a year or two of purchase. Some other robots, though, can turn out to be great investments that will help children discover joy in computational thinking for years to come.

While in some settings it is acceptable to buy an educational device that is likely to only work for a short period of time, most educators must be more judicious when selecting and purchasing these robots. The process of learning about, selecting, purchasing, and using these robots is an investment that needs to return a reasonable number of hours of use [6]. The longevity of these smart tangibles impacts more than just budgets; teachers and students invest significant time and emotions into robot initiatives [7]. Frustrating experiences do not help schools promote CS to their students, especially for those who are underrepresented in CS. Robots that cease to function—colloquially referred to as "bricked"—only contribute more e-waste to the world. As sophisticated educational technologies continue to evolve, it is important for educators, administrators, and other supporters of science education to be able to discern good investments from the bad. When faced with choosing between over 80 different devices (with more hitting the market every year), how does an educator know which is which? What questions should an educator ask when evaluating different products? What qualities should they look for to minimize the risks they are most vulnerable to? This chapter will explore the importance of long-term support in robots and will present a framework to assist educators in answering those questions.

The literature review will begin by discussing how other areas of computer science deal with similar problems of long-term support.

In part one of the findings section, examples of the problem with CS Ed robots will be explored. Part two will describe examples where a device was bought in a sealed package but was either unusable or had greatly reduced capabilities.

Part three of the findings section will introduce a proposed framework for educators to use when evaluating a potential CS Ed robot purchase and what kinds, and degrees, of risk may be incurred. An example of the framework being applied to a device is provided.

In limitations and future work, we will lay out a vision for further development of the framework so both consumers and designers of educational robots can move towards a future where robots are, if not everlasting, at least reasonably durable.

Finally, although this chapter approaches this problem through the lens of computer science, the analysis and framework can be applied to educational robots in other fields.

Issues around reliance on companion software, narrow compatibility with hardware, difficult to replace parts, and trust in companies with little history are not unique to CS Ed robots, and any educator considering purchasing an educational robot should be aware of these potential issues and apply the provided framework on a potential purchase.

2. Literature review

To motivate the need to address the problem of long-term use and stability in CS educational robots, we start by looking at the related issues as described in the literature.

2.1 Long-term support in computer science

Long-term support (LTS) for software is essential to the longevity of a product. LTS starts when software has all intended features released and is considered complete. During this phase, there will be no more additional features, and updates are limited to security and bug fixes. Security updates are necessary for internet-connected devices as flaws are identified after the product has been released and people have had time to analyze the software for defects. However, while some software is released with LTS, for most software it is not added. Software that is widely used or used by a major investor of the software's parent company is more likely to get an LTS stage.

A great implementation of LTS is the Ubuntu operating system's kernel. Ubuntu takes the approach of releasing new editions of their Linux kernel every 9 months and an LTS version of their kernel every 2 years. Ubuntu defines LTS as enterprise-focused, compatible with new hardware, and prioritizing testing over-development [8]. When a kernel is released as an LTS version, there will not be more features released and it will not use unstable packages or cutting edge software elements. The LTS is well documented for users: in the Ubuntu Wiki, there is a specific page on LTS defining exactly what LTS means to them and the standards they adhere to, and a short definition of LTS so users know that the kernel is different than their usual 9-month release. Users who download Ubuntu LTS kernels have a guarantee that the software they are downloading will be supported for 5 years.

Long-term support is not just a commitment to software updates and fixes, but also a commitment that the software will be funded and supported during that time. Funding is necessary so the software will have adequate customer support and other technical documentation available. When an LTS version is released, it is a sign that the company is confident that the product will last for the LTS period and that they are willing to fully support it for the full duration of the product lifetime. That confidence is not present in CS Ed robots, which creates the possibility that a school may invest in a product hoping for 5 years of service only for the company to abandon the product several months later.

2.2 Software Escrow

Software Escrow is a way to preserve the original source code in the case the original developer can no longer continue support for their work. The contract is between a client and developer, with the escrow agent acting as the storage for code that the developer creates. They perform checks and verification of code and ensure all code is submitted properly [9].

Escrow is an additional layer of guarantee on top of a license to use proprietary software; it is treated as an insurance policy if the licensed code is abandoned [10]. Source code is defined on each escrow contract. It normally includes source code, libraries, test data, databases, and documentation. Any quality assurance needs are also defined in the contract. Various methods could be used to assure code quality, but they need to be defined by the client. It is not typically guaranteed with the provided source code they could continue the development of the codebase.

2.3 Schools' technology budget

The most obvious problem with CS educational robots which quickly or unexpectedly break is the cost [11]. Many K-12 schools already struggle with the onerous burdens of affording technology with average spending of $18,000 per school per year for computer upgrades alone [12]. CS Ed robots are most effectively and substantially integrated with classroom instruction when there are enough for a class set (as opposed to one or two devices used in self-directed free time), but with the average per-student price of a CS Ed robot being $140, a classroom set quickly runs into the thousands of dollars. The additional context of many free, high-quality options to learn CS (such as Scratch or code.org) creates a high burden of expected usable hours for educators to justify an investment in CS Ed robots. Schools which serve low-income and underserved populations are both most likely to benefit from the engagement of CS Ed robots and the least likely to be able to afford to invest in an unreliable product.

Any investment in new technology is made with the expectation that the new technology will last for a reasonable duration, typically at least a couple of years, and, in the case of hardware and tangibles, withstand the normal wear and tear students may inflict.

Furthermore, school technology is often funded by grants or one-time funding opportunities that pay strictly for the hardware and not any of the maintenance costs. This is a potential problem for tangibles because their moving parts mean they have a higher maintenance burden, and companion software can have subscription fees. The Unruly Splat tangible, for example, requires a subscription for the Splats to be used, the app to be accessed, and other customer service perks. CS Ed robots with apps that are free initially may not remain free. Therefore, while grants are often used to kick start the purchase of new CS Ed robots, ongoing costs for maintenance of devices that fall outside the normal computer lab IT support will be an issue [13]. Furthermore, CS Ed robots have software that is not updated regularly may require a school to keep older iPads or tablets supported. If money is not set aside for maintenance, then high-risk CS tangibles would not be an option, no matter the educational benefits.

2.4 Teacher's time

Another limited resource for schools that deserves protection is a teacher's time. It takes a substantial investment of teacher time to select, learn to operate, maintain, and prepare classroom activities for CS Ed robots. It is also difficult for teachers to get "hands-on" experience with a robot before purchasing it: researching options is time-consuming, complicated, and must be done outside of school hours. Integrating new tangibles depends on many factors, such as compatibility with their past experiences, complexity of the technology, and if they can try it out before committing [14].

While most robots are sold with some guidance on how to use them in an educational setting, teachers still need to become familiar with the device and customize it for their teaching style and classroom context. If a robot is usable for a limited length of time, it is a waste of teacher time.

Wasting teacher time can have long-term ramifications. The field of instructional technology and CS education has a long history of new technologies which are heavily promoted yet do not fulfill the promises made, which can contribute to teacher skepticism about educational technology [15]. Therefore, it is important that teachers and schools can plan their tangible investments to be worth the investment over time.

2.5 App security

For the many CS Ed robots which rely on apps, the discontinuation of product support can create serious security vulnerabilities. Risks include exposing personal data of children used to create the user accounts, revealing payment methods for any "in-app purchases," giving malicious actors access to a device's camera or speakers, or even allowing a malicious actor to hack the app such that harmful content is displayed [16, 17]. It can even create legal liability for schools that have an obligation to protect student data [18].

2.6 Toys and E-waste

CS Ed robots which break or become bricked are likely to become electronic waste, or e-waste, as repair options are typically non-existent. The issue of large amounts of e-waste being generated by prematurely obsolete technology already exists. To give an idea of the scale of this problem, in 2012 the global e-waste totaled around 45.6 million metric tonnes [19]; in 2019 that value totaled over 53.6 million metric tonnes (on average 7.3 kilograms per capita) [20].

Tangibles can create e-waste in two ways. First, if the toys are irreparable, then those that break will be thrown away and new toys will be purchased to replace them. Secondly, if the tangibles are not backward compatible, or a new generation fully replaces an old version, then the older generations of the tangibles become obsolete and must be thrown away. In addition to the tangible itself, additional technology that is required that becomes obsolete can become a source of e-waste.

2.7 Emotional attachment

Finally, one of the features which help CS Ed robots do a great job of teaching beginner CS topics is how students become emotionally attached to the robots (which actually can contribute to students' success at solving coding challenges [21]). Children can become very emotionally attached to their robots [22, 23]. When the devices stop working, it can create distress [24, 25].

3. Methodology

This study uses artifact analysis as a methodology for analyzing the various design and user experience factors related to robots and smart tangibles. Artifact analysis is a process by which an artifact is analyzed to understand the design philosophy of an artifact and study its users [26]. Designers may use this process to generate pristine ideas for future designs. As an active process, artifact analysis may require additional primary or secondary research to understand the full implications of a product's design or usage [26]. Researchers can use various inquiry routes to closely analyze an artifact's various aspects, such as material, spatial features, functionality, and interactivity. In the process of analysis, researchers explore the data to tell them more about the aspect they are interested in [27]. The advantage of conducting an artifact analysis is that it does not require human research participants while yielding useful insights. The disadvantage of using this form of analysis is that it does not lend itself well to considering other factors outside the design or usability of an artifact, such as some of the questions we ask in our risk analysis frameworks like company history, product history, or legacy support.

In this study, having motivated the need for a way for educators to assess a CS Ed robot's stability, we purchased two dozen different CS Ed robots and then examined them for design features that would cause them to stop working. We used this information, in combination with consulting several educators who have worked extensively with CS Ed robots and customer product reviews to design the Computer Science Risk Analysis Framework for Toys (CS RAFT). To validate that the framework was accurate, comprehensive, and easy to use, we then used the framework to analyze six smart tangibles, after which we re-designed and improved the framework. The robots and smart tangibles that we analyze are Learning Resources Code and Go Robot Mouse Activity Set, Makeblock Codey Rocky, Makeblock CodeyBot, Wonder Cue Robot, Wonder Dash Robot, and LEGO Mindstorms EV3. We included the Code and Go Robot Mouse Activity Set as a typical example in this chapter; the analyzes of the others were omitted for space.

4. Findings

When it comes to CS education products there are little to no LTS options available. The phrases "Computer Science Education" and "Long-term support" returned no relevant papers on the first five pages of Google Scholar. In particular, no relevant research has been found about how to create general plans for LTS for CS education tangibles.

4.1 Issues identified

We identified a variety of potential ways in which a CS Ed robot may be rendered inoperable before its expected end-of-life. The largest risk is for devices that require apps or external programs. If the companion software is no longer updated, the product servers are taken offline, or the software becomes unavailable because it is removed from the relevant app store, then the robot may cease to be useful. An example is the Jibo robot: after the company was sold, owners were initially told their devices would largely stop functioning as the servers were turned off. The company was subsequently sold again, and as of the writing of this chapter, users are waiting to find out more [28]. These events can occur suddenly due to a company closing or the device's manufacturer gearing up to promote a new product.

Another way the devices may become inoperable is if the software is given a large enough update that outpaces the computing power of the hardware (tablets, laptops, etc.) that the school has access to.

Some devices have subscription models on their software, which may be affordably priced when the school purchases them but can increase to unaffordable levels.

The next most likely risk to a CS Ed robot's long-term stability is having parts break. Some are not designed for realistic durability given children's use profile or are constructed of custom plastic parts and electronics which are difficult or impossible to replace once broken. Others have parts that are small and easy to lose, especially in a classroom environment. Some robots may be designed with constant adult supervision in mind, which schools with low faculty-to-student ratios cannot afford.

On a product-by-product basis, the hardware may have short-term support in terms of limited warranties, but for the many that require supplemental software or a mobile app, there is often little mention of any software warranty or support. The Sphero device is a good example of a typical warranty: a 1-year warranty for the hardware but none for the software or its availability. Others (such as the Ozobot Evo, Wowee Coji, Sphero, and OsmoBot) were found to have a hardware warranty for only 30–90 days after purchase, but no software warranty. The Ozobot Evo goes as far as stating in section 13 of their Terms of Service that they do not guarantee any of the results they advertise on their website [29]. These types of warranties are a poor fit for schools that plan curriculum far in advance and very well might not get the devices in front of students within 30 days of purchase. Often there is no mention of a development life cycle for the apps or other accompanying software and no ability to predict when (or if) updates and bug fixes will occur.

4.2 Examples of abandoned robots

We analyzed two CS Ed robots that succumbed to some of the issues above: Codeybot and Codie. Codeybot was made by Makeblock after a successful 2016 Kickstarter (raising almost $200,00) and was extensively covered by popular media. Codie (made by Codie Labs), which was designed through several startup weekends and won awards, was featured in numerous press outlets while raising $96,306 on Indigogo [30].

4.2.1 Codeybot

Codeybot was expressly marketed to teachers, offering them a discount and promising, "Are you an educator? We want to help you bring coding to your classroom!." The authors were able to purchase a Codeybot new inbox on Ebay. It would turn on, but the Android app was no longer available; out of the two iOS apps, the one that let you manually control the features was last updated in 2016, and the one to learn to program was updated in 2018. When the company was emailed, they said they were no longer supporting the app as the device is in "End of Life." The website documentation had broken images. The company had not yet responded to a question about what the end of life process was and what support was offered to current owners. The device had sturdy construction but an embedded battery which will likely be the cause of its eventual failure.

4.2.2 Codie

The authors were not able to obtain a Codie robot despite an extensive search. However, despite the Indigogo campaign's claim that "Codie is designed with durability in mind," we were able to assess that devices that have shipped had limited functionality and are now likely no longer usable [31]. Although the Indigogo campaign claimed in

December of 2015 that the app would be available soon, the company shuttered in August of 2016. It appears that a handful of the almost 700 backers received devices that did work, but the software did not have "play or create" modes. One comment on the campaign said that they had received an email from the company where the company claimed its future was dependent on selling a lot of Codies to schools. One of the last comments on the campaign was: "Would have been nice if they would at least release the code and build specs so we could build our own." While this is an extreme case of how CS Ed Robots can be negative experiences for educators, most CS Ed robots sold today are vulnerable to the issues which doomed the Codie.

4.3 CS RAFT (Computer Science Risk Analysis Framework for Toys)

We developed CS RAFT in order to assist educators to identify the risks associated with buying educational tangibles for their classrooms. The framework is divided into four major categories: Contents, Concepts, Company, and Community.

Each educator may have access to different resources, and risks that can be taken on by one educator can vary from those that can be taken on by another. As such, the framework does not aim to assign a single value for risk or make any recommendations regarding purchases; instead, it is designed to present an educator with the essential questions to answer in order to make an informed decision about the risks they are willing to take.

As an example, we analyzed the Code and Go Robot Mouse Activity Set from Learning Resources. This standalone robot does not require any additional software or hardware and therefore has no risk of software incompatibility or deprecation issues. Learning Resources is a well-established company with many products and has a low risk of dissolution. However, community feedback suggests issues with quality control on the wheels, especially over repeated use in a classroom, and a broken wheel requires the entire robot to be replaced. An educator on a grant-funded program that can afford many replacements for a shorter-term project may be willing to take on this risk, while another educator looking for a long-term toy to use as a supplement in their classroom may not.

4.4 Advice for designers

For new robots that are being developed, there should be considerations for the longevity of the device. New devices could be improved if there were plans for both software and hardware flexibility so that in the case the project is discontinued, there would still be a way for users to continue enjoying the device to its fullest potential. Reaching out to the target community and getting feedback from end-users could help generate a solid customer base for the robot to ensure the success of the product. Finally, we encourage CS Ed robot manufacturers to work towards standards for communicating with the devices such that software can be replaced if it becomes unavailable.

In addition, robots should be made as modular as possible, and replacement modules and parts should be available for purchase. As students use the robots in the classroom, not all robots will break in the same fashion. Rather than forcing educators to purchase an entirely new robot because of one broken wheel, giving them the option to replace just the wheel (perhaps with one 3D printed from provided specifications) and repair the robot themselves would be ideal. This will allow for less e-waste and long-term use in the classroom and ultimately more educators being willing to invest in devices.

5. Contributions

This research highlights and documents a new and growing problem facing K-12 CS educators: the unpredictable longevity of CS Ed robots. To address this problem, this chapter presents a novel framework, CS RAFT (Computer Science Risk Analysis Framework for Toys), which can aid educators in understanding the risk profile of CS Ed Robots when selecting them for purchase, and can be of use to robot designers and manufactures in developing more devices.

The CS RAFT framework can be applied to educational robots and toys outside of computer science. Tangibles in all facets of education are evolving to integrate more technologies like robots with complex microcontrollers, companion software, and augmented or virtual reality systems. Happy Atoms teaches students chemistry and physics through modeling atoms and compounds using plastic pieces that can then be scanned with a tablet running a companion application. The application offers more information on the scanned molecules, guided learning tools, and more. Miko 3 is an artificial intelligence-driven robot that engages kids with a variety of STEAM (Science, Technology, Engineering, Arts, and Math) topics through its catalog of games and tools. Tools like Happy Atoms and Miko 3 have similar concerns to the CS Ed robots discussed in this chapter and should be analyzed through the CS RAFT framework. Educators may ask themselves about the usability of the Happy Atoms models if the companion app stops being supported, or the cost incurred at a school of recurrent payments for the Miko Max premium subscription.

6. Conclusion

In conclusion, there are numerous and significant impacts of CS Ed robots becoming prematurely unusable, but as we find in other areas of computer science, software and hardware longevity is a problem that has a number of solutions. This problem extends to all other "smart" devices which educators purchase, and while there are legal frameworks available in some countries to give customers remediation, for now, there is little protection for consumers in the United States. Educators would benefit from being able to make purchases based on a more informed risk profile of CS Ed robots, and designers could do more to prevent this problem. The CS-RAFT framework presented in this chapter can assist anyone poised to make a significant investment in a CS Ed robot to look more critically at the qualities of the robot. Applying the framework can better inform the purchaser on the potential risks of investment and make smarter decisions about the choice, use, or quantity of the robot. We hope this work will motivate more research in this area and move us towards a future where CS Ed robots are confidently purchased by educators with the expectation that they will last long enough to validate their investment.

7. Limitations and future work

One limitation is the rapidly evolving space of how robots are used to teach computer science, of legal frameworks around "fitness for sale," and of norms and standards in manufacturing robots. It is possible, although unlikely, that a solution will naturally arise in one of those spaces between the writing of this chapter and its publication. Another limitation is the limited amount of user testing our framework could receive from teachers who use CS Ed robots; this was primarily caused by the pandemic both reducing the use of tangibles and its impact on teachers' available time to assist in research. In future work, we will partner with teachers to extensively use and give feedback for the next iteration of the CS RAFT.

Appendix A

A.1 Risk analysis framework

Use the framework below to guide your computer science toy risk analysis, and consider your organization's needs and available resources. A toy that has high-risk factors in one category may still be worth investment. For example, a toy produced by a new startup with little history may seem risky from a company perspective, but if the toy can be easily maintained by the user, long-term company support may not be necessary.

A.1.1 Contents

The contents category considers the hardware and software of the toy itself to assess the impact of long-term wear, lost or broken materials, and discontinued support. While self-contained, independent toys run the risk of being difficult to repair or replace, they are also immune to issues arising from the interfacing tablet or computer. Educators should consider the resources available to them when evaluating the weight of the focus points in this section.

A.1.1.1 Software flexibility

- What app or software does this toy need?
- This app or software must be online and/or connected to a server. ☐
- This app or software is open-source. ☐
- This app or software is third-party software (e.g. Arduino IDE). ☐

A.1.1.2 Hardware flexibility

- Additional technologies required to play with this toy:
    - Personal computer/laptop ☐
    - Tablet (e.g., iPad) ☐
    - Other (e.g., Infrared camera, light sensor) ☐
- List any specific requirements for the hardware required (e.g., Must be an iPad; no Android tablets)

- Batteries are easily replaceable □

A.1.1.3 At-risk parts

Wheels, buttons, and small supplemental parts are at high risk of being lost, broken, or worn down even under proper use.

- This toy has:
  - Wheels □
    a. How vital are the wheels to learning with the toy?
  - Buttons □
    a. How vital are the buttons to learning with the toy?
  - Supplemental parts □
    a.List any supplemental parts that are vital for learning with the toy.

A.1.1.4 Replaceability

Consider the at-risk parts checked in the section above.

- How much does this toy cost (per unit)?
- Does the company offer replacements for at-risk parts?
- Can any of the at-risk parts be substituted by another off-the-shelf item?
- Can any of the at-risk seem repairable by the user?
- What is the warranty on the product?

A.1.2 Concepts

The concepts category considers the learning objectives, overall design of the toy, and the nature of play while using the toy. Adopting a new robot into the classroom environment requires investment beyond the price of the toy, such as lesson planning and teacher training. Choosing products that share similar concepts with others on the market can mitigate some of the risks involved in those investments, as those similar toys can be interchanged with minimal edits to the lesson plan. While there may be benefits to using a highly specialized toy, purchasing robots that are extremely unique in their conceptual design pose a higher risk to the educator of having non-transferable content if support for the toy is discontinued.

A.1.2.1 Core concepts

- What computer science concept does this toy aim to teach?
- Is this toy a self-guided toy, or does it require teacher facilitation?

A.1.2.2 Interface and accessibility

- How do people interact with the toy (e.g., touch screen, voice commands)?
- Does it meet the accessibility needs of your audience?
- How much space and preparation does this toy need?
- How many people can share the toy at once while being effective?

A.1.2.3 Estimated effective playtime

- How much time does the toy need to be played with for fun and effective learning?
- This toy requires continued, long-term progress on the same toy. ☐

A.1.3 Company

The company category considers the history and nature of the company that produces the toy. While it is difficult to predict the future of a company's direction, trends within the company for previous versions of the toy or related toys can be useful for understanding the risk of committing to using the toy long-term. Reliable support from a well-established company can mitigate some of the uncertainty involved around buying specialized, delicate, or otherwise risk-carrying toys.

A.1.3.1 Company history

- When was this company established?
- This company makes products beyond computer science education toys. ☐

A.1.3.2 Product history

- When was this product released?
- When was the last time the software for the toy received an update?

A.1.3.3 Versions and variants

- Are there previous versions or variant versions (e.g., same toy, but branded with a popular movie franchise) of the toy?
- Are previous versions still supported and marketed?

- Are parts backward-compatible with previous versions?

A.1.3.4 Legacy support

- What is the company policy on supporting legacy products?
- Is there evidence of products becoming open-source once discontinued?

A.1.4 Community

The community category considers the relationship between the toy and its various users, and analysis in this section can inform the answers to other categories in the framework. Ratings and feedback from the community can be litmus tests for company responsiveness, and they can also inform the user of risks of breakage or loss. A rich community of educators supported by first and third-party resources such as lesson plans, video demonstrations, and professional development opportunities can suggest signs of a stable toy.

A.1.4.1 User feedback

- What are the ratings and reviews from other users?
- Record any common reasons for critical comments and concerns.
  - The company responds to these public comments. ☐

A.1.4.2 Community forums

- A public forum for this toy exists (official or otherwise) for educators to participate in.
  - This community is active (most recent activity is within the last month). ☐

A.1.4.3 First-party resources

- The company offers the following resources for educators:
  - FAQs ☐
  - Educator guides/lesson plans ☐
  - Educator training/professional development ☐
  - Video guides and demonstrations ☐
  - Other resources (list them):

A.1.4.4 Third-party resources

- There are the following resources created by other educators and users for this toy:
  - Educator guides/lesson plans ☐
  - Customized extensions or parts ☐
  - Video guides and demonstrations ☐
  - Student artifacts ☐
  - Other resources (list them):

Author details

Ishin Iwasaki[1†], Corben Roszak[1†], Parama Chaudhuri[2†], Katherine LaRue[1†] and Caroline D. Hardin[1*]

1 Western Washington University, Bellingham, USA

2 Indiana University Bloomington, Bloomington, USA

*Address all correspondence to: hardinc3@wwu.edu

† These authors contributed equally.

References

[1] Livingstone S, Haddon L. EU Kids Online: Final Report. London, UK: London School of Economics and Political Science; 2009

[2] Yadav A, Hong H, Stephenson C. Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. TechTrends. 2016;**60**(6): 565-568

[3] Freeman MJ. Advanced verbal computers in education. Educational Technology. 1975;**15**(5):58-60

[4] Resnick M, Ocko S, Papert S. Lego, logo, and design. Children's Environments Quarterly. Cambridge, MA: Harvard University Press's; 1988; 5(4):14-18

[5] Sullivan A, Bers MU. Dancing robots: Integrating art, music, and robotics in Singapore's early childhood centers. International Journal of Technology and Design Education. 2018;**28**(2):325-346

[6] Papadakis S, Vaiopoulou J, Sifaki E, Stamovlasis D, Kalogiannakis M, Vassilakis K. Factors That Hinder in-Service Teachers from Incorporating Educational Robotics into Their Daily or Future Teaching Practice. In: Proceedings of the 13th International Conference on Computer Supported Education. Virtual Event; 2021. p. 55-63

[7] Papadakis S, Vaiopoulou J, Sifaki E, Stamovlasis D, Kalogiannakis M. Attitudes towards the use of educational robotics: Exploring pre-service and in-service early childhood teacher profiles. Education Sciences. 2021;**11**(5):204

[8] Mahnke P. LTS—Ubuntu Wiki. Available from: https://wiki.ubuntu.com/LTS

[9] Weigl E, Binder J, Strodl S, Kolany B, Draws D, Rauber A. Framework for Automated Verification in Software Escrow, Proceedings of the 10th International Conference on Preservation of Digital Objects; 2013. pp. 95-103

[10] Gauberti A. Do you need to put in place an escrow agreement with respect to the software and/or source code that you license? 2020. Available from: https://crefovi.com/articles/do-you-need-to-put-in-place-an-escrow-agreement/

[11] Soule T, Heckendorn R. Cotsbots: computationally powerful, low-cost robots for computer science curriculums. Journal of Computing Sciences in Colleges. Oct 2011;**27**(1):180-187

[12] Ireh M. Budgeting and Funding School Technology: Essential Considerations. School Business Affairs. 76(7):18

[13] Bielefeldt T. Systemic Planning for Technology. Oregon School Study Council Bulletin. Jan 1997;**40**(2):1-32

[14] Dias LB. Integrating technology; Learning and Leading with Technology. 1999;**27**:10-13

[15] Cuban L. Oversold and Underused. Harvard University Press; 2009

[16] Yong S, Lindskog D, Ruhl R, Zavarsky P. Risk Mitigation Strategies for Mobile Wi-Fi Robot Toys from Online Pedophiles. In: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing. Boston, MA; 2011. p. 1220-1223

[17] Albuquerque OP, Fantinato M, Kelner J, de Albuquerque AP. Privacy in smart toys: Risks and proposed solutions. Electronic Commerce Research and Applications. 2020;**39**:100922

[18] Reyes I, Wijesekera P, Reardon J, Elazari Bar On A, Razaghpanah A, Vallina-Rodriguez N, et al. "Won't Somebody Think of the Children?" Examining COPPA Compliance at Scale. 2018. Available from: https://dspace.networks.imdea.org/handle/20.500.12761/551 [Accepted: 13 July 2021]

[19] Perkins DN, Brune Drisse MN, Nxele T, Sly PD. E-waste: A global hazard. Annals of Global Health. 2014; **80**(4):286

[20] Forti V, Baldé CP, Kuehr R, Bel G. The Global E-waste Monitor 2020. Bonn, Geneva and Rotterdam: United Nations University/United Nations Institute for Training and Research, International Telecommunication Union, and International Solid Waste Association; 2020. 120 p.

[21] You S, Robert L. Emotional Attachment, Performance, and Viability in Teams Collaborating with Embodied Physical Action (EPA) Robots. Rochester, NY: Social Science Research Network; 2017

[22] Weiss A, Wurhofer D, Tscheligi M. "I love this dog"—Children's emotional attachment to the robotic dog AIBO. International Journal of Social Robotics. 2009;**1**(3):243-248

[23] Huang L, Picart J, Gillan D. Toward a generalized model of human emotional attachment. Theoretical Issues in Ergonomics Science. 2021;**22**(2):178-199

[24] Gjersoe NL, Hall EL, Hood B. Children attribute mental lives to toys when they are emotionally attached to them. Cognitive Development. 2015;**34**: 28-38

[25] Carter EJ, Reig S, Tan XZ, Laput G, Rosenthal S, Steinfeld A. Death of a robot: Social media reactions and language usage when a robot stops operating. In: Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction. New York, NY, USA: Association for Computing Machinery; 2020. pp. 589-597. DOI: 10.1145/3319502.3374794

[26] Noah L. Artifact Analysis. Available from: http://dlrtoolkit.github.io/artifact-analysis/

[27] Given LM. Artifact analysis. In: The SAGE Encyclopedia of Qualitative Research Methods. Thousand Oaks: SAGE Publications. 2008. Available from: http://www.yanchukvladimir.com/docs/Library/Sage%20Encyclopedia%20of%20Qualitative%20Research%20Methods-%202008.pdf

[28] Bartneck C. Why do all social robots fail in the market? 2020. Available from: https://ir.canterbury.ac.nz/handle/10092/101172 [Accepted: 26 October 2020]

[29] Ozobot Terms of Use. Available from: https://ozobot.com/terms-of-use

[30] Introduction to Programming—mBlockly for Codeybot. Available from: https://www.makeblock.com/project/introduction-to-programming-mblockly-for-codeybot

[31] Codie—Cute Personal Robot That Makes Coding Fun. Available from: http://www.indiegogo.com/projects/1006485/fblk