Sustainability and Quality: icing on the cake

Coral Calero
Instituto de Tecnologías y Sistemas de la Información
University of Castilla-La Mancha
Ciudad Real, Spain
Coral.Calero@uclm.es

Manuel F. Bertoa
Departamento de Lenguajes y Ciencias de la Computación
University of Malaga
Malaga, Spain
bertoa@lcc.uma.es

Mª Ángeles Moraga
Instituto de Tecnologías y Sistemas de la Información
University of Castilla-La Mancha
Ciudad Real, Spain
MariaAngeles.Moraga@uclm.es

Abstract— **In the literature several definitions of sustainability and software sustainability can be found. However, we think none of them can be applied to the software development context. For that reason, in this poster we cook some existing definitions and we will propose a new one specific for the software development. From this definition we try to create a new recipe, discovering how to combine sustainability and quality. As a result several questions arise. From the result of answering these questions, next steps on our research will be defined.**

Index Terms—**Sustainability, Software Sustainability, Software Quality, Non-Functional Quality Requirements.**

I. What is Sustainability Software?

The most cited definition of sustainable development is defined in [1] as “meets the needs of the present without compromising the ability of future generations to satisfy their own needs.” According to [1], sustainable development needs to satisfy the requirements of the three dimensions of society, economy, and environment. A fourth dimension, human sustainability, is less present in the public discussion.

Naumann el al. in [2] define Green and Sustainable Software as software, whose direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software are minimal and/or which has a positive effect on sustainable development. Fundamentally, a software product has a direct impact related to resources and energy consumption during its production and use. Indirect impacts are (1) effects from the software product usage together with other processes and (2) long term systemic effects.

However, we have not found definitions on what is software development sustainability. As we think it is fundamental for our work, we will try to create a new definition.

And the first we need in order to do this is to think about how a software product can affect the sustainability.

Software products mainly affect the environment through resource consumption that occurs during its use and production. The most direct (and obvious) impact by a software product is energy consumption, but other resources may also have a negative impact on the sustainability of the software. In this sense, the use of the processor, increased needs of memory and disk storage, network utilization and bandwidth, potential relocation of software production and use, among others, are elements to take into account.

If we mix up these two ingredients (sustainability and resource consumption) and using as basis the UN report [1], we can define sustainable software development as follows:

Sustainable software development refers to a mode of software development in which resource use aims to meet product software needs while ensuring the sustainability of natural systems and the environment.

II. Which Are The Resources To Optimize (to take into account) In Software Development

Once we have the definition of sustainable software development, based on the proper usage of resources, we need to be clear which are these resources. In this way, we could take them into account during the development of a software product, improving their use, in order to increase the sustainability of the final product.

Recent systematic literature reviews [3, 4] have detected that most of the measures found in the literature related to software sustainability measurement, were defined to measure energy consumption, either directly or as a performance indicator.

Agarwal et al in [5] state that software performance depends on the basic computing resources like the CPU and the memory. Software demands for faster CPUs and increased memory, that derives in a mandatory computer upgrading. In this situation the biggest sufferer is the earth’s sustainability, as manufacturing and disposal equally contribute to the carbon footprint. Moreover, we need to evaluate the software products sustainability in terms of memory requirements and processor time. Other resources that appear in literature are network bandwidth and wireless technology because they influence the ecological footprint of a software product.

In [6] authors analyze the variation of the power consumption in different versions of a software product. As a result, authors were not able to relate power consumption with size metrics. They also arrived to the conclusion that power use was not just the CPU’s fault: disk and memory usage also matter.

A few studies evaluated the use of other types of resources that appear as a result of software product utilization: paper wastage, ink and toner costs, etc.

An open question is whether a software product can directly produce CO_2 emissions or another kind of pollution.

As we can see all the proposals focus on the resource utilization. So, it is clear that on the software development these resources must be taken into account and must be optimized in order to achieve a sustainable product.

From this reflection, we can also define the sustainability of a software product as the capacity of developing a software product in a sustainable manner.

III. Non-Functional Quality Requirements

Once software sustainability has been defined and the software resources to be taken into account from the sustainability perspective have been identified, the next step is to study how to address software sustainability (we use "software sustainability" with the meaning of "software product sustainability").

When a software product is developed it is necessary to specify the requirements that the product should satisfy. Software requirements can be classified into functional and non-functional requirements. The former should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs [7]. Therefore, the functional requirements are related to the “What” of a software product.

Non-functional requirements can be seen as requirements that constrain or set some quality attributes upon functionalities [8]. This means that non-functional quality requirements can be seen as the “How” of a software product.

Taking into account the software sustainability definition it is obvious that sustainability is a way to improve a software product, and then it must be related to the how and not to the what. So, based on the definitions of functional and non-functional requirements we believe that software sustainability is related to non-functional requirements.

And now? We are sure that sustainability, similarly to quality, is related to non-functional requirements but, how do we combine both? This is the question we will try to answer next.

IV. ISO/IEC 25000

ISO/IEC 25000 is a series of standards specific for System and software product Quality Requirements and Evaluation, namely SQuaRE [9]. In the ISO/IEC 25010 division, three quality models are defined [10]. In fig. 1 the targets of the quality models and the related entities are shown.

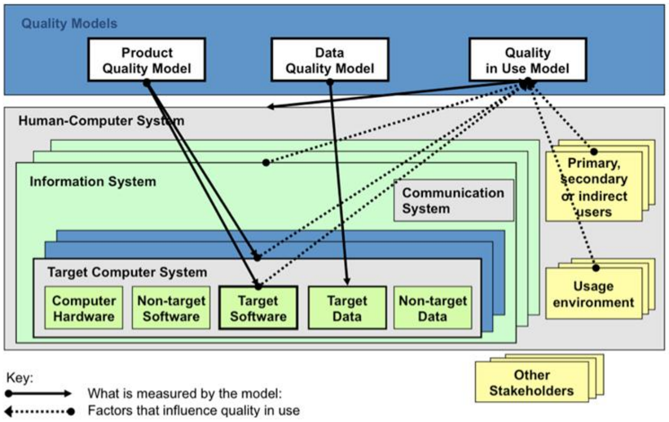


Fig. 1. Targets of quality model

From this figure we can observe that product quality is reflected on the Product quality model which is related to the target software.

The product quality model is composed of eight characteristics each one subdivided into several subcharacteristics (see fig. 2).

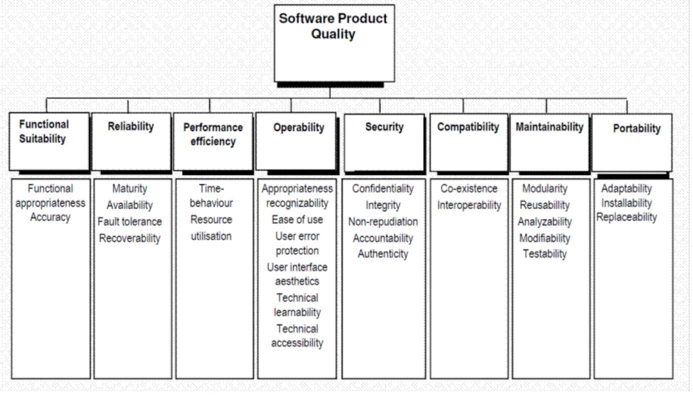


Fig. 2. Software product quality model in ISO/IEC 25010

Non-functional requirements reflect the quality necessities or desires for a software product. As sustainability is related to non-functional requirements, we think that sustainability must also be related (in some manner) to this product quality model.

V. Open Questions

A final conclusion about the question we are trying to answer about how to mix quality and sustainability, is that there are four possible ways to mix both. Unfortunately, these possibilities derive on new open questions that we need to discuss in order to decide which one is the good one:

Q1. Should be software sustainability considered as a new concept (at the same level of the software quality) having its own characteristics and subcharacteristics even sharing some of them with quality?

Q2. Should be software sustainability included as a new characteristic into the software quality model proposed by SQuaRE-25010?

Q3. Is the “performance efficiency” characteristic the same as the software product sustainability?

Q4. Is sustainability a way to achieve the quality, this means, can we have a sustainable quality and a non-sustainable quality?

The first option, considers the sustainability as an independent concept of quality. This means that sustainability must be considered, from all the perspectives, independently of quality. It will be necessary to think about sustainability, to define it, creating a complete model (similarly to how quality is defined) and explore how to work with it. Also it will be necessary to plan how to include the sustainability in the software development process.

On the second option, sustainability belongs to quality, being part of it. This means that sustainability must be considered as a new characteristic on the model. Then, all the operations and foundations given on the standard will be applied to the sustainability. In that case will be necessary to define

The third option, considers that sustainability is the same that the performance efficiency characteristic of ISO25010. This means that sustainability is already included on the standard. In that case the work must be focused on studying if really these two characteristics match (also their respective subcharacteristics). If we demonstrate this, then, no more work will be necessary.

Finally, the fourth option puts the sustainability as an influencing factor over the quality but no as a part of it. It is like, let's say, a way of doing things. The sustainability affects to each of the quality characteristics in the way that a quality characteristics can be achieved in a sustainable way or in a non-sustainable way. In this case, it will be necessary to define how to compose this new kind of interaction.

All the options present difficulties but can be considered as valid. However, choose the best option, or the correct one, will affect all the rest of our work because it will put the bases over which to follow the research on the achievement of sustainable software products.

So, considering fundamental to have the correct answer, we want to know opinions about it, beyond our own opinions and intuitions.

ACKNOWLEDGMENT

This work is part of the GEODAS-BC (TIN2012-37493-C03-01), the PEGASO (TIN2009-13718-C02-01) and Sistemas Inalámbricos de Gestión de Información Crítica (TIN2011-23795) projects funded by the Spanish Ministerio de Economía y Competitividad and by FEDER (Fondo Europeo de Desarrollo Regional).

REFERENCES

[1] United Nations World Commission on Environment and Development, "Report of the World Commission on Environment and Development: Our Common Future," In United Nations Conference on Environment and Development, 1987.

[2] S. Naumann, M. Dick, E. Kern, and T. Johann, "The greensoft model: A reference model for green and sustainable software and its engineering," Sustainable Computing: Informatics and Systems, vol. 1, (4) pp. 294-304, 2011.

[3] P. Bozzelli, Q. Gu, and P. Lago. A systematic literature review on green software metrics. Available at: http://www.s2group.cs.vu.nl/wp-content/uploads/2013/01/LS.PaoloBozzelli.GreenMetrics.pdf

[4] C. Calero, M. F. Bertoa, and M. Moraga, "A Systematic Literature Review for Software Sustainability Measures," in 2nd International Workshop on Green and Sustainable Software (GREENS 2013) in conjunction with ICSE 2013, 2013.

[5] S. Agarwal, A. Nath, and D. Chowdhury, "Sustainable Approaches and Good Practices in Green Software Engineering,".International Journal of Research and Reviews in Computer Science (IJRRCS), vol. 3, (1) 2012.

[6] A. Hindle, "Green mining: Investigating power consumption across versions.," in 34th International Conference on Software Engineering (ICSE), 2012, pp. 1301-1304.

[7] IEEE Std 830. IEEE Recommended Practice for Software Requirements Specifications. 1998.

[8] M. Glinz, "On non-functional requirements," in International Conference on Requirements Engineering, 2007, pp. 21-26.

[9] ISO/IEC 25000, "Systems and software engineering - Software product Quality Requirements and Evaluation SQuaRE," 2010.

[10] ISO/IEC 25010, "Systems and software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software product quality and system quality in use models.," 2010.

Sustainability and Quality: icing on the cake

Coral Calero
University of Castilla-La Mancha
Coral.Calero@uclm.es

Manuel F. Bertoa
University of Malaga
bertoa@lcc.uma.es

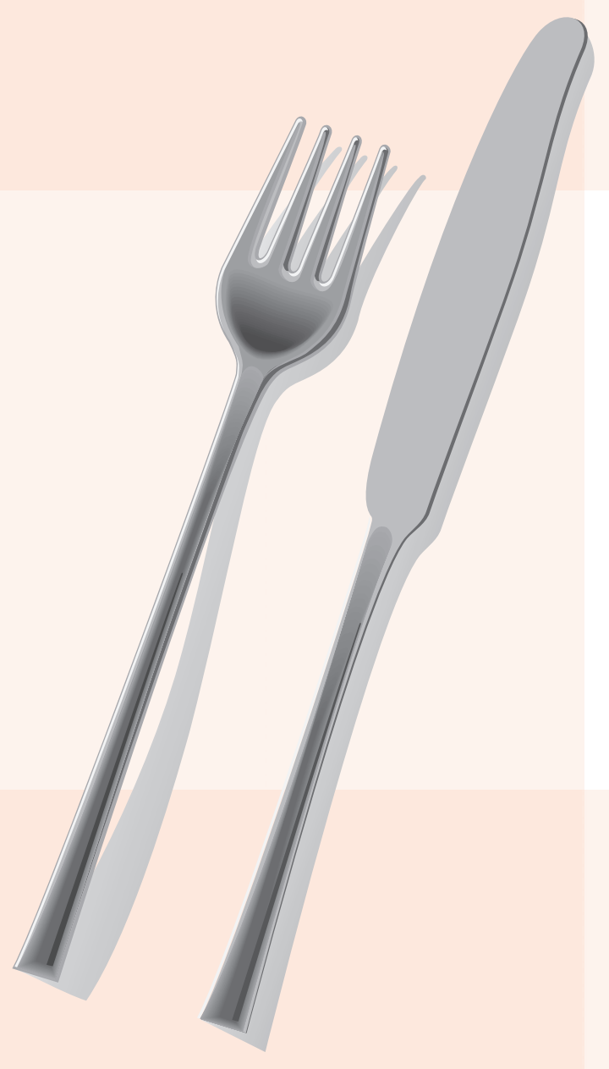
Mª Ángeles Moraga
University of Castilla-La Mancha
MariaAngeles.Moraga@uclm.es

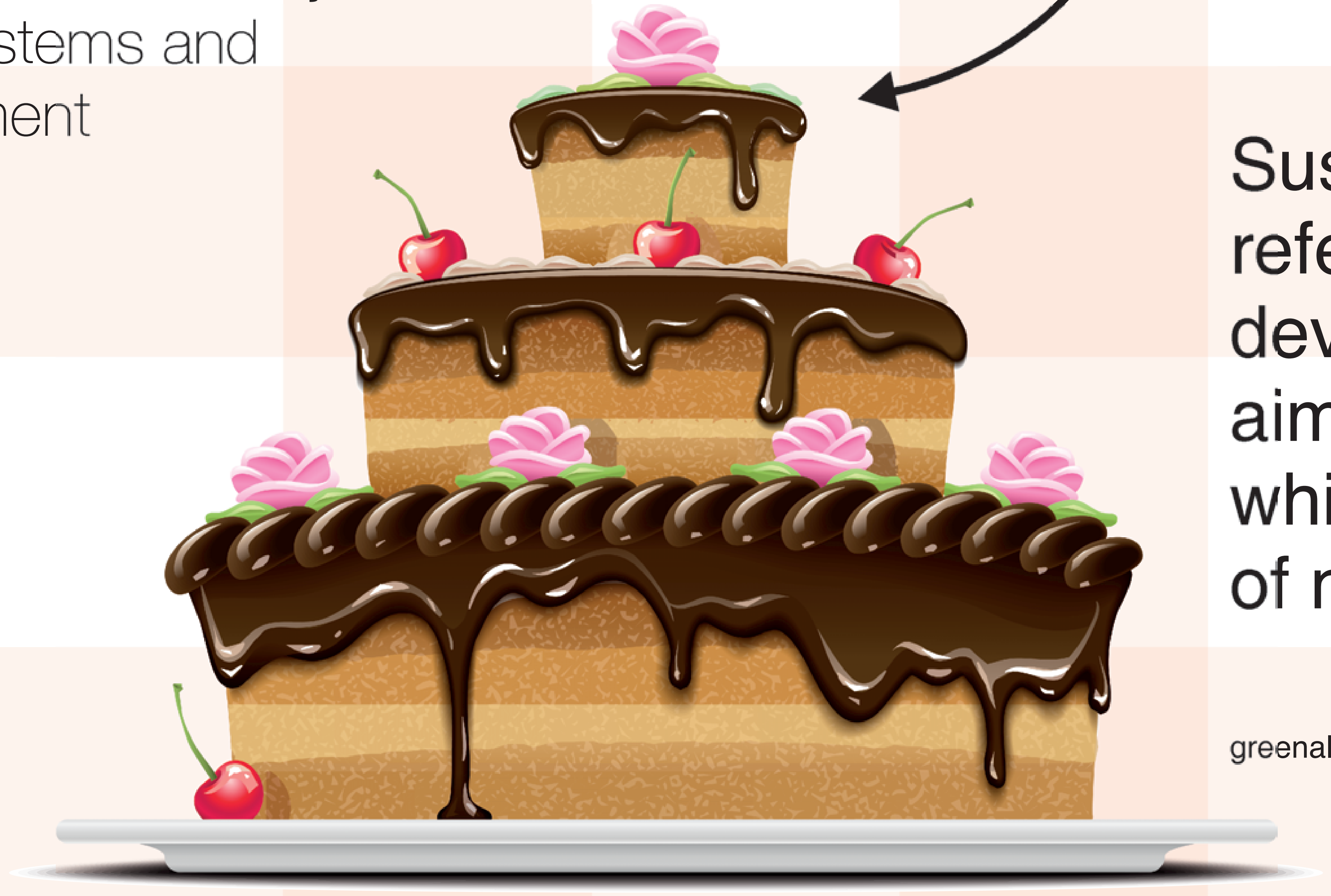
What is Sustainable development?

Sustainable development is a mode of human development in which resource use aims to meet human needs while ensuring the sustainability of natural systems and the environment

Which are the software resources?

Energetic consumption, Time consumption, hardware resources, other resources (paper, ink, ...).

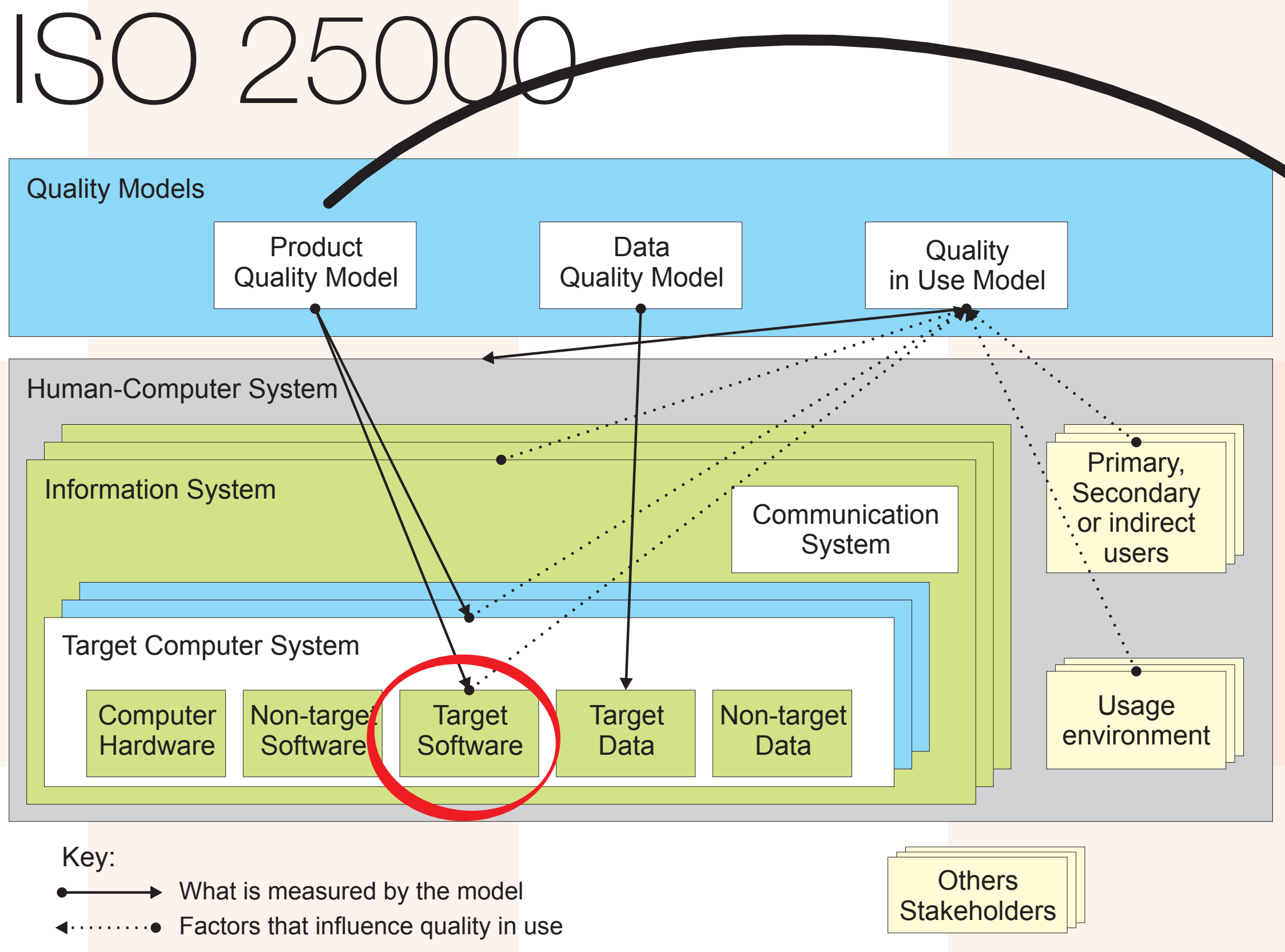




Sustainable software development refers to a mode of software development in which resource use aims to meet (product) software needs while ensuring the sustainability of natural systems and the environment.

greenalarcos.blogspot.com.es

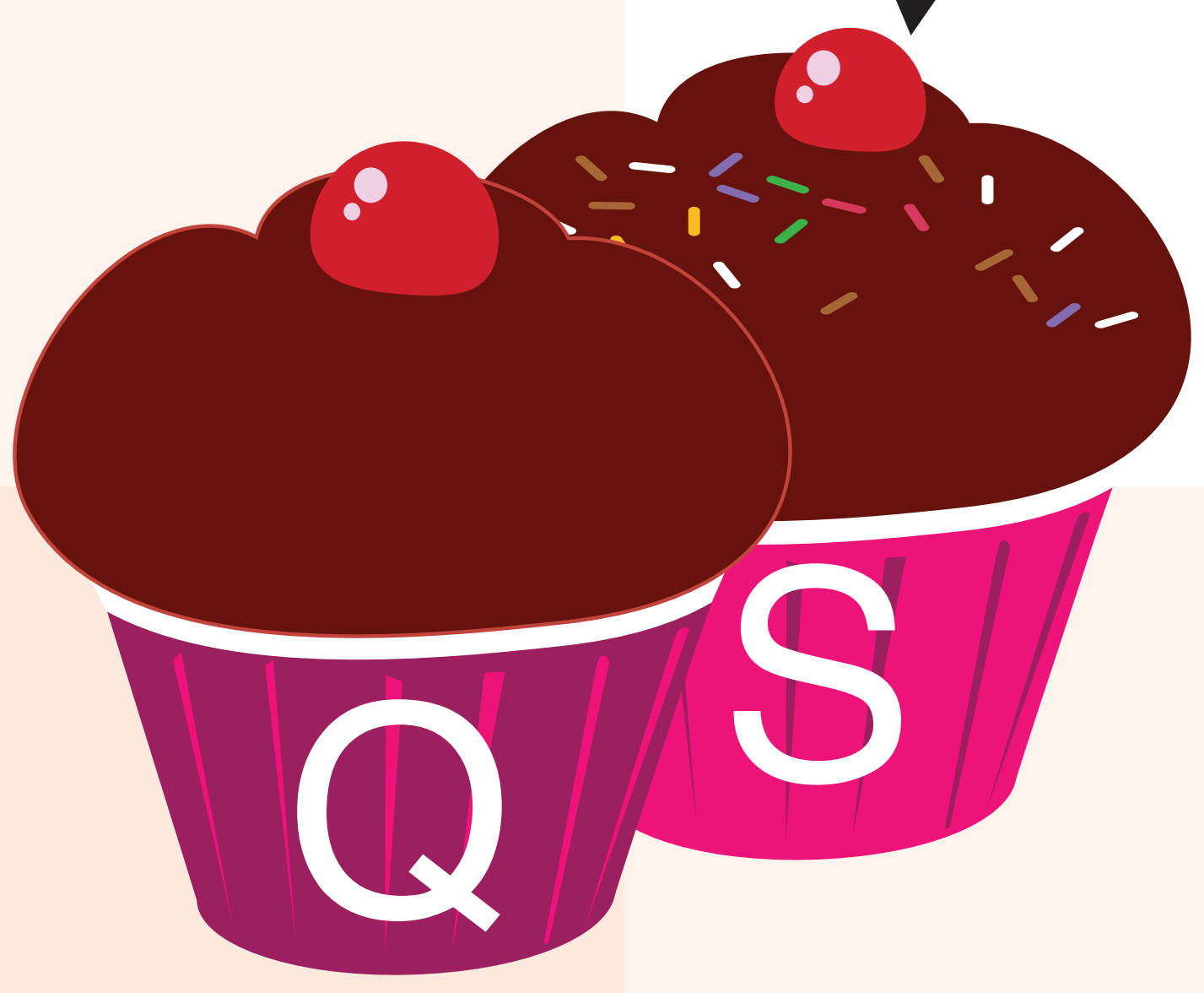
ISO 25000



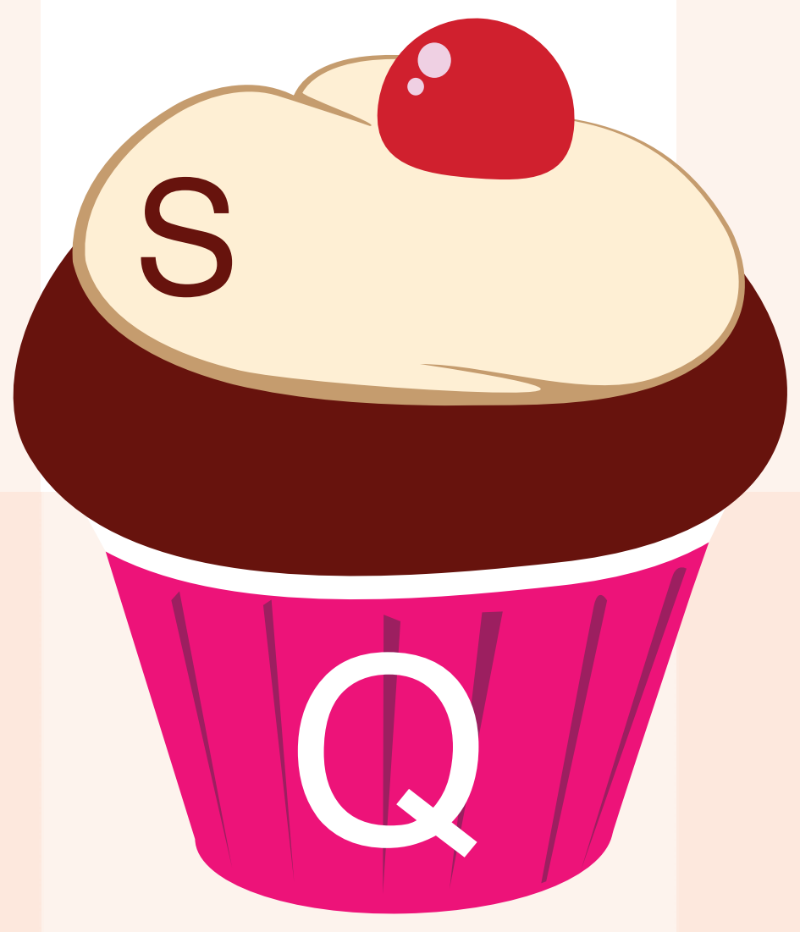
SOFTWARE DEVELOPMENT
WHAT
HOW

NON-FUNCTIONAL QUALITY REQUIREMENTS

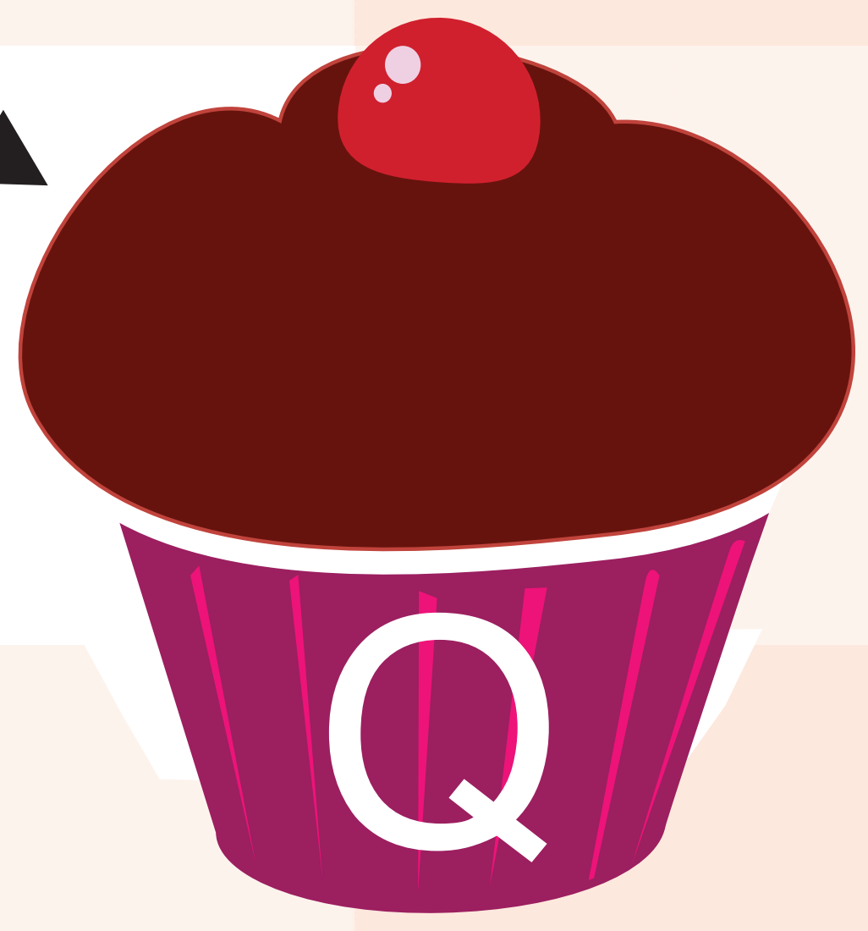
FUNCTIONAL REQUIREMENTS



DIFFERENT CONCEPTS THAT SHARE SOME CHARACTERISTICS



SUSTAINABILITY AS A PART OF QUALITY



PERFORMANCE EFFICIENCY THE SAME AS THE SOFTWARE SUSTAINABILITY

SUSTAINABLE QUALITY