

Chapter 22

Modeling Transparency in Software Systems for Distributed Work Groups

A B Sagar

Hyderabad Central University, India

ABSTRACT

Software systems require ethics. Several systems fail due to lack of ethics built into them. So, every software engineer needs to have an idea on ethics, so that they can build them into the software systems. Ethics requires accountability. Accountability is to the organization, stakeholders, and to the society. This chapter discusses the ethical issue of transparency. In the previous decade, we have seen a very increased use of this term. Transparency is being considered an indispensable ingredient in social accountability and is necessary for preserving and guaranteeing ethical and fair processes. Transparency is related to visibility of information, and without it, the organization and stakeholders will be left in blind states. Ultimately, the lack of transparency leads to unpleasant surprises due to bad decisions, letdown, increasing doubts, uncertainty, failure, and the breakdown of faith and trust. This is bad for the organizations involved, the people investing in them, stakeholders, suppliers, and employees. It is also bad, as we have seen recently, for the economy. This is true for government agencies and functions just as for businesses and communities. The least transparent governments are the most dictatorial and secretive in their decision-making and governance. An informed citizenship is vital for healthy, free societies, just as informed investors are needed for a healthy, strong economy. The degree of transparency in the software systems, then, becomes a barometer for health and vitality in governments, leadership, and business. The growing importance to the requirement of transparency in all these domains was the motivation to the present chapter.

DOI: 10.4018/978-1-4666-3679-8.ch022

1. INTRODUCTION

Transparency policies have proven to be effective for resolving controversial issues by minimizing health and safety risks, fighting corruption, promoting civil rights, sustaining improvements to public services, etc. Recent observations revealed that there is an increase in societal attention to the issue of transparency and it was also predicted that transparency will become the required premise for gaining and maintaining customer trust and collaborative relationships with all stakeholders in all constructive software systems. This chapter studies transparency and provides a basis for implementing transparency in a software system. The work presented in this chapter can be used in any software system to facilitate transparency into the system. The objective of this chapter is to provide a rudimentary framework for transparency and also provide a few implementation details to facilitate implementation of transparency in any software system.

Doubts and uncertainty can lead to loss of faith and trust. Loss of faith and trust on a software system, leads to collapse of the system. Transparency – which can be defined as “the accessibility of information to stakeholders of a business, regarding matters that affect their interests” – can shape and revolutionize a software system. Transparency policies have proven to be effective for resolving controversial issues by minimizing health and safety risks, fighting corruption, promoting civil rights, sustaining improvements to public services, etc. Recent observations revealed that there is an increase in societal attention to the issue of transparency and it was also predicted that transparency will become the required premise for gaining and maintaining customer trust and collaborative relationships with all stakeholders in all constructive software systems.

2. TRANSPARENCY

Transparency means openness of decisions and actions. Put another way, it means a free flow of information about decisions and actions, from source to recipient. There are several categories that are considered under transparency. Publication: The software system simply provides some rudimentary information about a specific issue; citizens/users can get a basic insight into operations and personnel; Transaction: The software system automates some public sector processes to remove partly those processes from human control; Reporting: The software system provides specific details of public sector decisions and actions; these may be reported in the form of performance indicators; Openness: The software system not only provides details of current performance, but enables users to compare that performance against pre-set standards (may also be called benchmarks or targets). Thus transparency is related to both financial and non-financial reporting. Financial reporting includes tracking of monetary data and non-financial reporting includes task execution data. This section provides a theoretical basis for understanding transparency and the conclusion provides details needed for implementation of transparency in a work group.

A work group implies two or more individuals who routinely function like a team, and interdependent in achievement of a common goal, and may or may not work next to one another or in the same department. This kind of work groups are ever present in business domains or as software teams. They represent a part of a business or the business itself. Transparency implies visibility of information related to financial and non-financial matters of the work group and its stakeholders. Work groups are generally hierarchical in nature with one supervising over another i.e. higher

10 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/modeling-transparency-software-systems-distributed/75759

Related Content

Agile, Lean, and Service-Oriented Development, Continuum, or Chasm

Juha Rikkilä (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 132-163).

www.irma-international.org/chapter/agile-lean-service-oriented-development/77703

Testing E-Learning Websites

Kamaljeet Sandhu (2012). *Advanced Automated Software Testing: Frameworks for Refined Practice* (pp. 182-195).

www.irma-international.org/chapter/testing-learning-websites/62156

Analysis of Cloud and Self-Web-Hosting Services Based on Security Parameters

Surbhi Khare and Abhishek Badholia (2022). *International Journal of Information System Modeling and Design* (pp. 1-14).

www.irma-international.org/article/analysis-of-cloud-and-self-web-hosting-services-based-on-security-parameters/297629

Exploring the Perceived End-Product Quality in Software-Developing Organizations

Jussi Kasurinen, Ossi Taipale, Jari Vanhanen and Kari Smolander (2012). *International Journal of Information System Modeling and Design* (pp. 1-32).

www.irma-international.org/article/exploring-perceived-end-product-quality/65560

Breast Cancer Prediction and Control Using BiLSTM and Two-Dimensional Convolutional Neural Network

Moses A. Agana, Chukwuemeka Odi Agwu and Nsinem A. Ukpoho (2023). *International Journal of Software Innovation* (pp. 1-19).

www.irma-international.org/article/breast-cancer-prediction-and-control-using-bilstm-and-two-dimensional-convolutional-neural-network/316169