

Chapter 17

The Software Principles of Design for Data Modeling: Gathering Requirements – A Detailed Guide

Saurabh Bhattacharya

 <https://orcid.org/0000-0002-2729-1835>

Chitkara University, India

ABSTRACT

The method for researchers to ascertain clients' software needs is requirement gathering. Requirement gathering rarely occurs successfully, and numerous software programs have failed as a result of incorrect or partial knowledge of the needs of users. The requirement-gathering technique is widely considered to be an essential part of development. The chapter will critically discuss why many developments fail due to poor requirements gathering. It is a challenging task to elicit requirements. When requirements are elicited, errors are most common. Addressing the system's needs presents several challenges, and the chapter intends to study these challenges and provide a solution. Professionals face challenges in gathering requirements due to the unavailability of stakeholders, unclear requirements, frequent changes in demand, and lack of skills for analysts. In a variety of contexts and areas, interviews with a preference for framework proved to be among the best gathering approaches.

INTRODUCTION

Information systems design places a high priority on gathering needs from users as well as other stakeholders. As any problems with the collected requirement have an effect on the operation as an entirety as well as certain circumstances that might result in a project's being unsuccessful, "requirements gathering" in software design is a crucial component of every "project". Software development is "the process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design

DOI: 10.4018/978-1-6684-9809-5.ch017

The Software Principles of Design for Data Modeling

in code, testing the code, and sometimes, installing and checking out the software for operational use” This term is significant as it describes the software as a malleable reality made up of various artifacts, including: “technology programs, processes, and perhaps related information and material related to the functioning of a hardware network.” The caliber of software development is influenced by the architectural choices made. A collection of solutions called “software design principles” aids in the creation of successful computer designs. The technique of drafting or creating a narrative exhibits many similarities with that of creating software, which has gained increased recognition in current decades. Every software creation venture must start with “requirements gathering”. They include statements such as “What the User wants to do?” How does it occur? In actuality, “requirements gathering” is the act of compiling an array of needs (functional, systemic, technological, etc.) from all the contributors (clients, users, suppliers, IT personnel), which will serve as the foundation for the official description of the features that the work involves. Inadequately drafted requirements may trigger issues throughout design since they characterize the venture, and most importantly, they can lead to programs failing if the objectives are unclear, in which case the project will fail. In keeping with the saying, “failing to plan, and you plan to fail”. A crucial step in the design of information technology is determining the requirements, however, there is strong proof that the procedure can and ought to be enhanced. Amendments can result in considerable advantages for the whole systems construction procedure since the majority of the required identification happens at the beginning of the building phase of an architecture(Browne & Ramesh, 2002). The essential stage in every software creation approach is “requirements gathering” for projects. Every undertaking must-have requirements to be defined, estimated and managed. All software development projects must adhere to high standards to be successful, no matter if it relies on “Agile Methodology” or the conventional “Waterfall Approach”. While conventional frameworks and agile techniques might employ distinct ‘requirements gathering’ procedures, the standards for good requirements are always the same(Davey & R. Parker, 2015).

WHAT IS REQUIREMENT GATHERING?

A project’s features are all determined through an approach known as “requirements gathering”, sometimes known as “requirements elicitation”. Technological and business requirements constitute the two basic categories of project specifications. The technical specifications describe how the project must be carried out whereas business criteria specify what an organization will achieve with the project. The method of determining the precise requirements for the project from beginning to end is known as requirements gathering. While managing the needs over the project timeframe, this process takes place throughout the project’s beginning stage. Obtain, accumulate, retrieve, or discover are all definitions of the word “Gathering.” Feature elicitation, in the context of software design, is the procedure used to collect specifications from key players for the creation of software. The stage of demand-gathering information kicks off every software creation project. As part of the “requirement gathering” method, analysts record the requirements, “wants”, and “desires” of stakeholders using a variety of gathering approaches. The analyst focuses on comprehending the needs, goals, and limitations of the mechanism that will be developed throughout the requirement-gathering process. Further specifications are defined during the specification elicitation step, which determines the requirements’ scope. Any development of an Application starts with an understanding of the “Wants” of the user and then moves to the “Desire” of the user. The first and ongoing stage of creating a “software application” is specification collecting.

20 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/the-software-principles-of-design-for-data-modeling/330499

Related Content

Simple System Dynamics and Control System Project Models

A. S. White (2014). *Systems and Software Development, Modeling, and Analysis: New Perspectives and Methodologies* (pp. 113-133).

www.irma-international.org/chapter/simple-system-dynamics-and-control-system-project-models/108812

An Analysis on Task Migration Strategy of Big Data Streaming Storm Computing Framework for Distributed Processing

Xiling Hu (2020). *International Journal of Information System Modeling and Design* (pp. 18-35).

www.irma-international.org/article/an-analysis-on-task-migration-strategy-of-big-data-streaming-storm-computing-framework-for-distributed-processing/265547

A Study on Major Service Items of Consumers and Companies Using Convergence Technology in the Intelligent Age

Chang Hwa Baek (2022). *International Journal of Software Innovation* (pp. 1-12).

www.irma-international.org/article/a-study-on-major-service-items-of-consumers-and-companies-using-convergence-technology-in-the-intelligent-age/301220

CbSSDF: A Two-Layer Conceptual Graph Approach to Web Services Description and Composition – A Scenario Based Solution Analysis and Comparison with OWL-S

Xiaofeng Du, Malcolm Munro and William Song (2011). *International Journal of Information System Modeling and Design* (pp. 82-103).

www.irma-international.org/article/cbssdf-two-layer-conceptual-graph/58646

A Machine Learning-Based Framework for Diagnosis of Breast Cancer

Ravi Kumar Sachdeva and Priyanka Bathla (2022). *International Journal of Software Innovation* (pp. 1-11).

www.irma-international.org/article/a-machine-learning-based-framework-for-diagnosis-of-breast-cancer/301221