Journal of Software Engineering and Applications, 2023, 16, 622-639
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Insights into B2C E-Commerce Quality Using ISO 25010

Antonia Stefani, Eleni Vamvatsikou, Bill Vassiliadis

Department of Management Science and Technology, University of Patras, Patras, Greece
Email: astefani@upatras.gr

How to cite this paper: Stefani, A., Vamvatsikou, E. and Vassiliadis, B. (2023) Insights into B2C E-Commerce Quality Using ISO 25010. *Journal of Software Engineering and Applications*, **16**, 622-639.
https://doi.org/10.4236/jsea.2023.1611032

Received: September 25, 2023
Accepted: November 27, 2023
Published: November 30, 2023

Abstract

This paper introduces an innovative Software Quality Assurance framework tailored for B2C e-commerce systems, seamlessly integrating software quality with business objectives. Drawing from elements of the ISO 25000 series and ISO 20000 standards, this framework specifically addresses challenges inherent to e-commerce. By establishing business-relevant KPIs, the framework ensures that ongoing improvement initiatives resonate with the company's strategic goals. Additionally, the paper presents a Dynamic Bayesian Network model as a hands-on tool for implementing the framework within e-commerce organisations.

Keywords

Electronic Commerce, Software Quality, Business to Consumer, Targeted Software Design, ISO 25010

1. Introduction

Quality is a business advantage in a globalised economy. The quality of the software supporting online business has long been recognised as such [1]. Significant advancements have been made in measuring software quality, albeit the methods and tools do not formally integrate business parameters such as goals and objectives [2] [3]. The need for comprehensive frameworks in software quality assurance research is driven by the increasing complexity of software systems and the growing demand for software products that fulfil business objectives [4]. Business software is more complex and interconnected, especially the one that enables business-to-consumer (B2C) processes. It embraces inter-organisational processes, suppliers and customers to serve the enterprise's business objectives. As competition becomes fierce in a globalised digital economy, the need to find competitive advantages depends on the extent to which the company's policy is successfully implemented by the information systems software it uses, and particularly on how well the software aligns with the processes, not only on a technical level but also in a business sense [5] [6].

Software quality standards, such as ISO/IEC 25000 series, focus on several aspects of software assessment and have proven excellent yet general guidelines for, mostly, assessing than designing quality software. These standards do not explicitly consider the company's business objectives in implementing or operating the software. By design, they are generic enough to be flexible and applicable to various software and use cases. Since business objectives can vary greatly from one economic sector to another, the relationship between the software and the business goals remains highly indirect [7].

We argue that there are gaps to be filled in as far as software quality assessment of business-to-consumer e-commerce systems is concerned in the sense that software quality standards do not explicitly consider business objectives. Quality-driven design of such systems could use business and quality objectives to achieve focused design, redesign or continuous development utilising an economy of scale model. Business objectives are (or should be) central to any system's design. Goals such as boosting sales and augmenting user engagement to elevate operational efficiency are hidden in user requirements at the analysis stage of a system's lifecycle. It's imperative that the design inherently supports and strives to achieve these goals more dynamically.

Furthermore, quality objectives like reliability, performance, security, and usability are intertwined with many businesses' aims. By embedding business and quality benchmarks into the design process, we boost the design of systems that align with business aspirations, come closer to optimal user experience, and possibly achieve high levels of operational resilience. Achieving a sufficient alignment between business and quality objectives is difficult.

This work proposes a new Software Quality Assurance (SQA) framework for B2C e-commerce systems by expanding the work in [8] [9]. A B2C service or system is decomposed into components—units of functionality, the most fundamental, non-overlapping functions the system provides users. These components are (ideally) supplementary, and their aggregation constitutes the system's functionality. Theoretically, each component can be developed, tested, updated, or replaced independently. Thus, components also vary in their quality attributes and quality assessment of individual components becomes essential, especially when considering a system holistically. This approach ensures that the system functions optimally and can adapt to changing requirements or conditions over time. Supplementing the technological view of the system/service, user views of the software are introduced, Facets. Facets offer a unique perspective when considering software systems, especially from a user-centric standpoint. Instead of focusing solely on the technical underpinnings, as with components, facets emphasise the holistic user experience, encompassing a set of functions that seamlessly merge to provide the user with a coherent and meaningful interaction.

Components are decomposed (mapped) into Facets and to a business view of the system represented by Key Performance Indicators (KPIs). KPIs are used to measure the success of the software in meeting these objectives. Having Components mapped to Facets and KPIs achieves a focused yet multi-view of how the software is used. Components are related to ISO/IEC 25010 quality characteristics (ISO/IEC 25010 being part of the ISO/IEC 25000 series standard), another application of the "divide and conquer method" which provides focused quality assessment. The practical use of the framework is exhibited by a prediction model that enables forward (from business to software design goals) and backwards (from quality to business goals) evaluation. A Bayesian Network is a probabilistic tool for depicting the prediction model.

This research contributes to the state of the art of B2C e-commerce quality by introducing a novel Software Quality Assurance framework that holistically merges software quality standards with business objectives, a fusion that is largely unexplored. It is one of the first attempts to formally use the ISO 25000 series and the ISO 20000 standards in conjunction. A cornerstone of this research, the Dynamic Bayesian Network model, not only amplifies the framework's practicality but also offers e-commerce organizations a tangible, data-driven tool to facilitate targeted design (and re-design) considering not only technical but business objectives as well.

The paper is structured as follows: first, the theoretical background on ISO standards for software quality is reviewed, and its determinants are summarised. In Section 3, the Quality Assurance Framework is presented, where the quality parameters of the model are underlined. Section 4 presents an in-depth description of the B2C Dynamic Bayesian Network in the application analysis section. Finally, the practical and theoretical implications of the quality improvement approach are outlined in the conclusion.

2. State of the Art

Several software quality assurance frameworks have been proposed in recent years to address some or all the challenges of software development and quality assurance. In the last decade, the International Standards Organisation (ISO) has published some standards that, albeit general, may be suited to manage the Quality Assurance process of modern software systems. These frameworks differ in their approach, scope, and level of detail, each having advantages and disadvantages. They have been praised for their flexibility and criticised for their somewhat lack of applicability. No single, unified framework is suitable for all types of software development. Some frameworks address specific domains, such as B2C e-commerce or mobile health applications; others provide a more general approach to software quality assurance.

ISO/IEC 25000 series and ISO/IEC 20000 are international standards that provide guidelines and frameworks for assessing the quality of software services and service delivery, respectively [10]. These two standards differ in their approach, scope, and level of detail since service delivery (supported by ISO/IEC 20000) focuses on the processes and alignment with business needs. In contrast, product quality (supported by ISO 25000 series) focuses on the attributes of the software product itself. Both are crucial for ensuring (directly or indirectly) that IT aligns with business objectives and delivers organisational value. The still theoretical idea of integrating components from ISO/IEC 20000 with components of ISO/IEC 25000 via a comprehensive framework may offer new advantages to organisations, namely serving more efficient business needs through high-quality software products, better stakeholder communication, holistic decision-making, and improvement opportunities.

The ISO/IEC 25000 series (or SQuaRE) is a suite of standards focusing on systems and software quality requirements and evaluation. Like all formal standards of the International Standards Organization, it provides a structured, consistent approach for specifying, measuring, and evaluating software quality attributes. The main components of the ISO/IEC 25000 series include ISO/IEC 25010, which describes the quality model and ISO/IEC 25020, which provides guidelines for developing quality measures and their evaluation (**Table 1**). Its implementation to e-commerce applications can be complex and time-consuming: standards by default have broad scope, lack explicit application guidelines and have non-existing upper and lower boundaries (backed by sectorial surveys) for practical use of their metrics.

ISO/IEC 20000 is an international standard for IT service management, not software, per se [10]. However, it emphasises continuous improvement and enhances customer satisfaction by ensuring the delivery of reliable and efficient software services. Its structure is organised into processes and sub-processes (since it is a process standard), but at the bottom level, it uses KPIs (Key Performance Indicators) instead of metrics (**Table 2**).

KPIs are indeed a fundamental component of the ISO/IEC 20000 framework. They offer the means required to assess, to some degree, that software services align with corporate goals and satisfy client needs. The ISO/IEC 20000 KPIs focus on IT Service Management, with the standard offering guidelines on factors that should be considered while creating and utilising the KPIs rather than providing specific ones. This somewhat reduces its practicality since the standard user needs to have a rather high level of understanding of both the framework's mechanisms and how KPIs express the business goals.

The design process of an e-commerce system, or even its expansion, upgrade or integration with other systems, may benefit from the standard despite these disadvantages. Alignment to business objectives and meeting customer needs are requirements that are more easily understood and translated into software design parameters. Continuous improvement is a more difficult goal to grasp and translate into system design specifications. As far as e-commerce software quality is concerned, we are interested in two characteristics: 1) the definition of KPIs for each process where one or more components deliver each process of the

Table 1. The ISO/IEC 25010 structure for product quality assessment (ISO/IEC 25000 series).

ISO/IEC 25010 Characteristic	ISO/IEC 25010 Sub-Characteristic
Functional Suitability	• Functional Completeness • Functional Correctness • Functional Appropriateness
Performance Efficiency	• Time Behaviour • Resource Utilisation • Capacity
Compatibility	• Co-Existence • Interoperability
Usability	• Appropriateness Recognisability • Learnability • Operability • User Error Protection • User Interface Aesthetics • Accessibility
Reliability	• Maturity • Availability • Fault Tolerance • Recoverability
Security	• Confidentiality • Integrity • Non-Repudiation • Accountability • Authenticity
Maintainability	• Modularity • Reusability • Analysability • Modifiability • Testability
Portability	• Adaptability • Installability • Replaceability

software and 2) continues improvement through the measurement of KPIs to identify areas for improvement and implement necessary changes, not only of processes but also of software components, indirectly. ISO/IEC 2000 does not provide the link (or better, the mapping) of KPIs to software quality characteristics (e.g. as defined by the ISO/IEC 25000 series).

Considering the above discussion, the research question is how exactly the two standards, ISO/IEC 20000 and ISO/IEC 25000 series, can be combined. ISO or the literature does not provide a specific combined framework or set of guidelines for the joint implementation of these two standards at any level. Instead, there is a general directive to organisations that encourages them to integrate the

Table 2. The ISO 20000 structure for service delivery.

ISO/IEC 20000 Requirements and Processes	ISO/IEC 20000 Processes
Service Management System General Requirements	• Management Responsibility • Governance of Processes Operated by Other Parties • Documentation Management • Resource Management
Design and Transition of New or Changed Services	• No Sub-Processes Defined
Service Delivery Processes	• Service Level Management • Service Reporting • Service Continuity and Availability Management • Budgeting and Accounting for Services • Capacity Management • Information Security Management
Relationship Processes	• Business Relationship Management • Supplier Management
Resolution Processes	• Incident and Service Request Management • Problem Management
Control Processes	• Configuration Management • Change Management • Release and Deployment Management

principles and practices of both standards into their existing processes. The how and what remain undefined. To this end, ideally, an Integrated Management System (IMS) should be defined. This unified system integrates all organisational processes, standards, and certifications into one framework. The IMS should ideally combine service management and software quality processes into a single system to achieve greater efficiency, consistency, and operational effectiveness. This is a high-level approach to what needs to be done.

3. The Quality Assurance Framework

3.1. Concept

This work proposes a customisable framework for software quality based on ISO 25010. The structure preserves a number of core, "stable" aspects of ISO 25010. Customisable components are introduced so that the framework can be adapted to reflect the business objectives of the organisation using the software.

Stable Components (SC) depict aspects of software quality as defined by ISO 25010. These are the eight characteristics of Product Quality, namely:

1) Functional Suitability;
2) Performance Efficiency;
3) Compatibility;
4) Usability;
5) Reliability;

6) Security;
7) Maintainability, and;
8) Portability.

Additionally, five characteristics of Quality in Use, namely:

1) Effectiveness;
2) Efficiency;
3) Satisfaction;
4) Freedom from Risk, and;
5) Context Coverage.

are also included to provide a more solid foundation for assessing software quality.

Customisable Components (CC) depict the business objectives. For each organisation, there could be a set of additional software quality characteristics that are specifically related to their unique business goals (and thus mapped to KPIs). These customisable components are linked to the core ISO 25010 aspects, and they are supplementing them. This approach would also allow for evolution over time, supporting continuous development. The customisable components of the software quality framework can be updated as the business objectives of an organisation change to reflect these changes.

A deep understanding of software quality principles and the specific business context is required to implement such a customisable framework. A higher level, customisable and flexible approach benefits this approach. To this end, this work expands on the concepts introduced in [8] [9], where a quality model was introduced based on the ISO 9126 standard (currently replaced by the ISO 25000 series). Overall, an e-commerce application was decomposed into components that were mapped to facets levels and quality sub-characteristics of ISO 9126. This divide-and-conquer strategy allowed us to synthesise the quality of the whole system based on the weighted quality of its components. And since each component was of a different quality, "texture", it has to be measured (in terms of quality) differently.

More specifically, ISO 9126 served as a foundation for decomposing quality into hierarchical levels and aspects, which, in turn, were further divided into sub-characteristics. The overall quality of an e-commerce application was conceptualised using three interrelated aspects that collectively offered a comprehensive functional depiction of the system: navigation, presentation, and purchasing (the Facets). The suggested methodology emphasised the examination of each aspect by breaking them down into components, which are the most fundamental, non-overlapping functions provided by the system to its users. These components were supplementary; their aggregation constitutes the system's total functionality. Moreover, components for the quality characteristics of ISO 9126—Functionality, Usability, Efficiency, and Reliability—were organised into Levels. The proposed model comprised two levels: Level 1 was defined by components independent of the underlying support technology, and Level 2 of components that may evolve (in response to technological advancements). Using field experts or prior knowledge, a weight calculation method for each component was also offered as a practical tool for targeted, quality-driven system design.

3.2. Structure

The proposed QAF retains the notion of divide and conquer but expands the structure significantly by using ISO 25010 instead of ISO 9126, using more Facets and introducing KPIs instead of technical complexity levels. The structure (based on the divide and conquer methodology) is depicted in **Figure 1**. The overall system (its quality) is decomposed into Components mapped to Facets and KPIs. The mapping functions may use weights that measure the significance given to the relation of a component with a Facet or KPIs. Facets are mapped to ISO 25010 quality characteristics and sub characteristics. The mapping can also be weighted depending on the focus the designers would like to pose on specific aspects of software quality. The decomposition of components (mapping to Facets and a business view of the system represented by Key Performance Indicators, or KPIs) is thus introduced. KPIs are used to assess how well the software has accomplished these goals. Components being mapped to Facets and KPIs results in a concentrated yet comprehensive understanding of how the product is used.

The following section provides a more in-depth view of the QAF.

3.2.1. Components

The foundational building blocks of the software are the components. Together, they make up the entire software architecture, with each component representing a discrete unit of functionality. The methodology of breaking a system down into these smaller parts helps ensure that developers can assess their quality more easily (for functioning components) or set quality-driven design goals (for components under development).

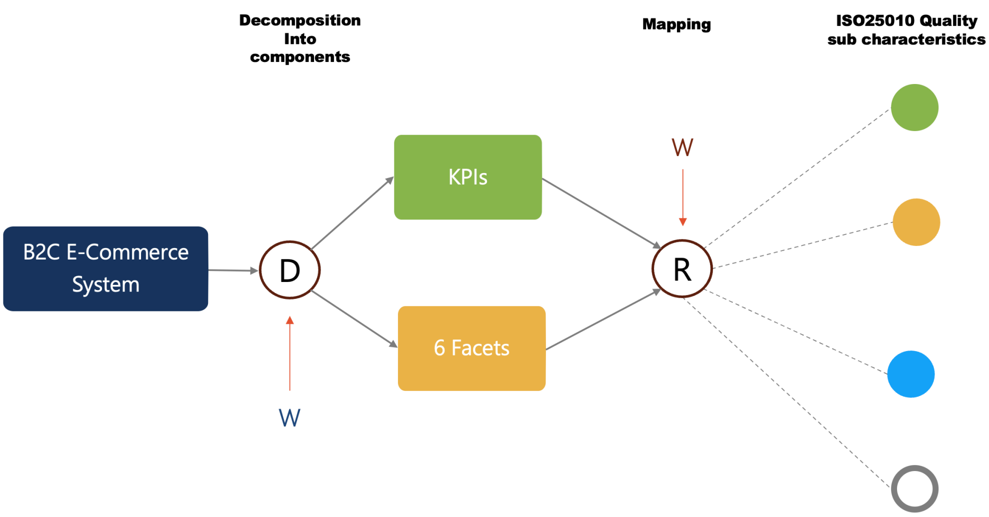

Figure 1. The "divide and conquer" methodology of the framework.

An e-commerce platform's components make the technical details of the platform's operation clearer. There are components dedicated solely to handling user authentication (ensuring users can register, log in, and maintain their sessions securely). Other components focus entirely on the product catalogue, managing product listings, prices, descriptions, and images. Other components handle payment processing, interfacing with various payment services to ensure secure transactions. In an actual e-commerce system, each component serves a different purpose. However, they are all made to work together seamlessly.

3.2.2. Facets

A facet is a conceptual tool in system design which bridges the gap between technical architecture and user experience. User experience facets are the customer journey using functions to perform a complete action. For example, the purchasing facet is the user's experience in using the system's functions from selection to checkout, the browsing experience as they navigate through different product categories, view details, browse through customer comments, etc. Each facet represents a comprehensive journey through the system from a user's perspective.

Facets are included in the QAF to underscore that software needs to be more than just technically sound; it must be usable. User experience is a quality parameter and a driver for achieving business goals. A single facet may be mapped to several components (simple system functions). For example, the purchasing facet relies on the shopping cart component for adding products, the product catalogue component for product details, the user authentication component for user details, and the payment processing component for completing the transaction.

Based on the above discussion, the proposed software quality assurance framework for B2C e-commerce applications can be expanded to include, besides navigation, purchasing and presentation, the following additional facets:

1) Customer Service—this facet describes the mechanisms provided to the end user for accessing customer service and support, including email, phone, chat, and social media support.

2) Order Fulfilment—this facet describes managing and fulfilling customer orders, including tracking and updating order status, handling cancellations and refunds, and generating shipping labels.

3) Marketing and Advertising—this facet describes the mechanisms used to promote products and the brand, including email marketing, social media marketing, search engine optimisation, and pay-per-click advertising.

Thus, the total number of Facets the QAF uses is six (6).

3.2.3. KPIs

KPIs are derived from ISO/IEC 20000 and integrated into the QAF. They are specific for organisations using B2C software. The methodology is simple, yet it requires a deep understanding of what is to be measured and how it contributes to business goals. The first step is that understanding the organisation's objectives and how the B2C software aligns with these goals. This alignment ensures the derived KPIs are relevant and contribute to the broader business strategy. Next, a thorough assessment of the current state of the organisation's IT service management processes is conducted. This involves evaluating how well the organisation meets the requirements of ISO/IEC 20000. By identifying gaps or areas of improvement, the organisation may pinpoint which KPIs are most needed to monitor and measure performance effectively. Once a clear understanding of the current state has been established, KPIs may be tailored specifically to the organisation's needs. These KPIs should be designed to measure the effectiveness and efficiency of IT service management processes.

Once the KPIs are designed, the next step is to connect them with the ISO/IEC 25010 software quality criteria. This entails completing a thorough software quality assessment, concentrating on usability, dependability, and performance. By comparing the results of this assessment to the established KPIs, the organisation gets a comprehensive picture of both service delivery and software quality. Following identifying areas for improvement, the organisation can also take focused activities to improve service delivery and software quality. By mapping KPIs from ISO/IEC 20000 to ISO/IEC 25010 characteristics, the QAF bridges the gap between service management and software quality.

KPIs that best represent the user experience from that perspective are set for each identified Facet. Only strong relations are recorded. This involves understanding each Facet's unique attributes and needs and selecting KPIs that can effectively measure how well the software meets those needs. The goal is to ensure that for every Facet, corresponding KPIs offer a clear and measurable view of the software's contribution to business goals. Since the identification and mapping of KPIs largely depend on the characteristics of the organisation using the QAF, an exhaustive mapping is not practical. However, **Table 3** provides an example of how a mapping of Facets to Quality Characteristics to KPIs may appear.

It is important to elaborate on the practical use of such a mapping to derive the appropriate KPIs. For example, the "Incident Resolution Time" KPI represents the average time taken to resolve incidents users report. This KPI is part of the Incident and Service Request Management process of ISO/IEC 20000. A sudden increase in this KPI might indicate potential usability issues with the platform, as users might encounter delays or errors. These may lead to dropouts and financial loss.

On the other hand, ISO/IEC 25010 defines "Usability" as a primary software quality characteristic. This characteristic focuses on aspects such as user error protection, ease of learning for new users, and the visual appeal of the user interface. By integrating the insights from the Incident Resolution Time KPI with a usability assessment based on ISO/IEC 25010, the organisation can comprehensively understand service delivery and software quality. For instance, if many incidents are related to users struggling with the checkout process, it could point to a usability challenge in that workflow.

Table 3. Example: KPIs mapped to ISO 25010 characteristics and QAF facets.

Facet	ISO/IEC 25010 Characteristic	KPIs
Navigation	• Functionality • Usability	Service availability impact on user retention, conversion rate influenced by navigation efficiency, revenue impact due to navigation failures, customer drop-off rate during search operations, churn rate linked to navigation complexities, average session duration affected by navigation design, sales impact of efficient search functionalities, return rate due to navigation misunderstandings, customer support tickets related to navigation issues, user engagement rate with intuitive navigation, feedback volume on navigation experience, cost of navigation-related incidents, impact of failed searches on sales funnel drop-offs, revenue impact of efficient filtering and sorting, customer lifetime value influenced by navigation ease.
Presentation	• Functionality • Usability	Service availability impact on sales, conversion rate based on content presentation, revenue impact due to presentation failures, customer retention rate post content updates, churn rate linked to content presentation issues, average order value influenced by product information clarity, sales impact of video and image load times, return rate due to product information discrepancies, customer support tickets related to content misunderstanding, user engagement rate with presented content, feedback volume on content presentation, cost of presentation-related incidents, impact of link failures on sales funnel drop-offs, revenue impact of content accessibility compliance, customer lifetime value influence by content presentation quality.
Purchasing	• Functionality • Usability	Service availability impact on purchase completion, conversion rate affected by purchase process efficiency, revenue loss due to order management failures, customer drop-off rate at payment gateways, churn rate linked to purchase process complexities, sales impact from seamless order management, return rate influenced by payment process clarity, customer support tickets related to purchase issues, user engagement rate with streamlined checkout, feedback volume on purchase experience, cost of purchase-related service incidents, impact of payment gateway downtimes on sales, revenue correlation with order tracking functionality, customer lifetime value affected by purchase experience ease.

4. Predicting Quality with Bayesian Networks

The QAF can be used for probabilistic reasoning through a Dynamic Bayesian Network (DBN) for forward prediction (future estimation) and backward assessment. The goal here is to rank the components of an e-commerce system based on their perceived significance by the designer (quality) or the organisation strategists (business goals). DBNs have already been utilised successfully in the modelling and production of software processes with good results [9].

The practical application of the model is the most important part of the evaluation process, including, as in [8] [9], a forward and a backward use. In the forward use, values for the KPIs are inserted, and the DBN can then be used to provide estimations about the system quality and the corresponding probability values. In the backward use, the DBN computes probabilistic values for an e-commerce system's developer-centred external measurements. These external metrics are inextricably linked to the internal development phase measures. By setting the quality design goals (through metrics that correspond to quality sub-characteristics), it is possible to predict the values of the KPIs, that is, which business goals are met.

4.1. The QAF Dynamic Bayesian Network

A DBN is a probabilistic graphical model representing the conditional dependencies between random variables. It uses a directed acyclic graph. In the context of the QAF, the KPIs, the Facets, the quality characteristics, and the quality sub-characteristics of ISO 25010 (and the metrics, if included) could be represented as random variables in a Bayesian network. The edges between the nodes indicate the conditional dependencies between them. A DBN is particularly useful in an e-commerce application where several factors may change rapidly over time.

In the graph model of the DBN, the parents are the KPIs, the sons of the KPIs are the Facets, the sons of the facets are the quality characteristics, and the leaves are the quality characteristics. The internal nodes of the tree have weights that represent the relative importance of each facet or quality characteristic. Metrics can be used at the lower level. The graph is depicted in **Figure 2**.

Let us examine each node in more detail:

- KPI nodes: These nodes represent the root of the DBN and have no incoming edges. Each KPI node has a probability distribution over its possible values, which would be learned from the data.
- Facet nodes: These nodes represent the intermediate level of the DBN and have incoming edges from the KPI nodes and outgoing edges to the quality characteristic nodes. The probability distribution for a facet node depends on its parent KPI nodes' values and weight, which would be learned from the data.
- Quality characteristic nodes: These nodes represent the next level of the DBN and have incoming edges from the facet nodes and outgoing edges to the quality sub-characteristic nodes. The probability distribution for a quality characteristic node depends on the values of its parent facet node(s) and its weight, which would be learned from the data.

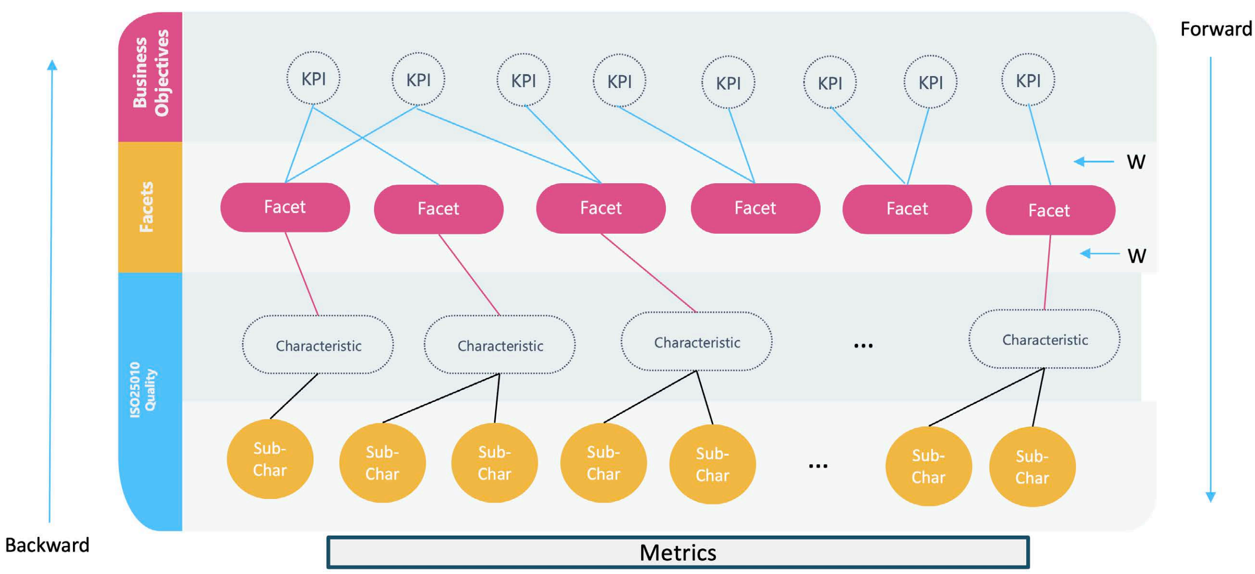

Figure 2. The DBN graph of the QAF.

- Quality sub-characteristic nodes: These nodes represent the leaf level of the DBN and have incoming edges from the quality characteristic nodes. The probability distribution for a quality sub-characteristic node depends on the values of its parent quality characteristic node(s) and its weight, which would be learned from the data.

Several types of weights can be used in the DBN to model the relative importance of different nodes in the tree, such as:

- Conditional probabilities: These weights reflect the likelihood of a specific value for a node given the values of its parent nodes. In the context of the QAF DBN, this would mean assigning probabilities to each node based on the probability of a certain value given the values of its parent nodes.
- Expert judgments: These weights reflect the subjective opinions of domain experts on the importance of different nodes in the tree. In the context of the QAF DBN, this could mean asking experts to assign weights to each node based on their experience and knowledge.
- Data-driven weights: These weights reflect the importance of different nodes in the tree based on their impact on the quality sub-characteristic values. In the context of the QAF DBN, this could mean using statistical methods to analyse the relationship between the KPIs and quality sub-characteristics and assigning weights based on the strength of the relationship.

The choice of weights depends on the specific context and whether data are available. It is essential to select weights that accurately reflect the relative importance of different tree nodes and lead to accurate predictions of the quality sub-characteristic values.

4.2. Calculating Probabilities

The KPI nodes represent the observed data and have conditional probability distributions (CPDs) based on the values of the observed KPIs. The facet nodes have CPDs based on the weights assigned to each KPI contributing to that Facet. The quality characteristic nodes have CPDs based on the weights assigned to each facet contributing to that quality characteristic. Finally, the sub-characteristic nodes have CPDs based on the weights assigned to each quality characteristic contributing to that sub-characteristic. The probability of the facet is calculated as follows:

$$P\left(Facet \mid KPI_1,\ KPI_2,\cdots,KPI_k\right)=\exp\left(w_1KPI+w_2KPI_2+\cdots+w_kKPI_k\right)/Z$$

where w_1, w_2, ..., w_k are the weights assigned to each KPI contributing to the facet, and Z is the normalising constant that ensures the probabilities sum to 1.

The formula for the conditional probability distribution of a quality characteristic node given its parent facet nodes is as follows:

$$\begin{aligned}&P\left(QC \mid Facet_1,\ Facet_2,\cdots,\ Facet_k\right)\\&=\exp\left(w_1Facet_1+w_2Facet_2+\cdots+w_kFacet_k\right)/Z\end{aligned}$$

where w_1, w_2, ..., w_k are the weights assigned to each Facet contributing to the quality characteristic, Z is the normalising constant.

Finally, the formula for the conditional probability distribution of a sub-characteristic node given its parent quality characteristic nodes would be:

$$P(Sub \mid QC_1, QC_2, \cdots, QC_k) = \exp(w_1 QC_1 + w_2 QC_2 + \cdots + w_k QC_k) / Z$$

where w_1, w_2, ..., and w_k are the weights assigned to each quality characteristic that contributes to the sub-characteristic, and Z is the normalising constant.

It is possible to use the DBN to make predictions about the values of the quality characteristics given specific values of the KPIs. By updating the KPI nodes with observed data, the probabilities of the other nodes in the network could be inferred using the forward-backwards algorithm or different inference algorithms. In addition, the internal nodes (Facets and quality characteristics) may have weights that represent their relative importance in the DBN graph. These weights affect the likelihood of specific paths being taken through the tree and help model the KPIs' impact on the quality characteristics.

4.3. The Forward-Backward Algorithm

Let X_t represent the KPIs at time t, $F_{j,t}$ represents the jth facet at time t, $C_{k,j,t}$ represent the kth quality characteristic of the j-th facet at time t, and $Q_{l,k,j,t}$ represent the l-th quality sub characteristic of the k-th quality characteristic of the j-th facet at time t. The conditional probabilities for each type of node are:

- For KPI nodes:

$$P(X_t \mid X_{t-1})$$

which is the probability of KPIs at time t given the KPIs at time $t-1$.

- For Facet nodes:

$$P(F_{j,t} \mid X_t)$$

which is the probability of the j-th facet at time t given the KPIs at time t.

- for Quality characteristic nodes:

$$P(C_{k,j,t} \mid F_{j,t})$$

which is the probability of the kth quality characteristic of the j-th facet at time, t given the j-th facet at time t.

- for Quality sub characteristic nodes:

$$P(Q_{l,k,j,t} \mid C_{k,j,t})$$

which is the probability of the l-th quality sub characteristic of the k-th quality characteristic of the j-th facet at time t given the k-th quality characteristic of the j-th facet at time t.

The weights for the internal nodes can be incorporated into the conditional probabilities as follows:

- For Facet nodes:

$$P(F_{j,t} \mid X_t) = \exp(w_j^f * X_t) / \sum_{j=1}^{m} \exp(w_j^f * X_t)$$

where w_j^f is the weight for the *j*th facet.

- for Quality characteristic nodes:

$$P\left(C_{k,j,t} \mid F_{j,t}\right) = \exp\left(w_{k,j}^c * F_{j,t}\right) \Big/ \sum_{k=1}^{n} \exp\left(w_{k,j}^c * F_{j,t}\right)$$

where $w_{k,j}^c$ is the weight for the *k*th quality characteristic of the *j*th facet.

- for Quality subcharacteristic nodes:

$$P\left(Q_{l,k,j,t} \mid C_{k,j,t}\right) = \exp\left(w_{l,k,j}^q * C_{k,j,t}\right) \Big/ \sum_{l=1}^{p} \exp\left(w_{l,k,j}^q * C_{k,j,t}\right)$$

where $w_{l,k,j}^q$ is the weight for the *l*th quality sub characteristic of the *k*th quality characteristic of the *j*th facet.

To predict the value of a quality sub characteristic $Q_{l,k,j,t}$ at time *t*, given a set of KPIs X_t, we use the Forward Algorithm to compute the joint probability distribution of all the nodes in the network at time *t*. Then, we use the marginal distribution of the desired node to predict as follows:

$$\begin{aligned} P\left(Q_{l,k,j,t} \mid X_t\right) = \sum_{F_{1,t},\cdots,F_{m,t}} \sum_{C_{1,j,t},\cdots,C_{n,j,t}} P\left(F_{1,t} \mid X_t\right) \cdots P\left(F_{m,t} \mid X_t\right) \\ \cdot P\left(C_{1,j,t} \mid F_{j,t}\right) \cdots P\left(C_{n,j,t} \mid F_{j,t}\right) P\left(Q_{l,k,j,t} \mid C_{k,j,t}\right) \end{aligned}$$

where the summations are overall possible values of the facet and quality characteristic nodes at time *t*, the weights can be learned from data mining techniques.

4.4. An Application Scenario

The DBN is used to assess the system's quality and predict future performance fo a B2C platform of a company. Assessment is focused on the understanding of the significance of various components of the e-commerce system, especially from the perspectives of design quality and business goals.

In the Forward Use of the DBN, the values for the KPIs are inserted. The KPIs used are related to the system's presentation facet, such as "service availability impact on user retention" and "conversion rate influenced by navigation efficiency".

The DBN is used to obtain estimations about the system's quality and the corresponding probability values. For instance, if the service availability is high, the DBN might predict a high user retention rate with a certain probability.

In the Backward Use, the quality design goals related to the presentation facet are set. Particular goals may include a user-friendly layout and providing clear product information. The DBN computes probabilistic values for developer-centered external measurements, such as the expected load time for product images or the accuracy of product descriptions. By setting these quality design goals, the DBN predicts the values of the KPIs, indicating which business goals are likely to be met.

5. Discussion

This research shows room for enhancement to current software quality standards to more directly involve business objectives [11]. The inclusion of business objectives presents its own challenges: business objectives can vary significantly from one organisation to another, making it difficult to develop a generic set of criteria that would apply to all software. It is difficult to measure the alignment of software with business objectives, a complex task with increased subjectivity that foremost requires a deep understanding of the business context [12].

Despite these obstacles, a stronger emphasis on business goals within software quality standards may offer organisations a valuable foundation for ensuring that their software is technically sound and strategically aligned with their business objectives. The Quality Assurance Framework presented in this work incorporates critical performance indicators that align with the business's goals and objectives. By setting KPIs relevant to the company, the framework can ensure that continuous improvement efforts are aligned with the business's strategic objectives. A Dynamic Bayesian Network model was provided as a practical tool for applying the framework to e-commerce organisations.

The DBN has some drawbacks, primarily in predicting the proper weights established by experts (a human factor) or context-sensitive data. These edges have weights or probabilities attached to them to show the relationships' strength. However, certain restrictions exist when choosing the correct weights for the BDN. The danger of human bias increases when relying on expert knowledge, which might result in subjectivity and inconsistent weights. Various perspectives or various levels of experience among experts could result in inaccurate predictions in the final model. Expert assessments might also understate the intricacy of the connections between variables, which would cause the model to be oversimplified. Although the data-driven methodology is more objective, it relies heavily on the data's accuracy and representativeness. The calculated weights might not effectively reflect the relationships between variables if the data is sparse, noisy, or impacted by context (such as hidden variables, temporal dynamics, or other factors not included in the model). Overfitting is another issue, mainly when the sample size is tiny in comparison to the complexity of the model. Overfitting occurs when the model gets overly specialised for the training set of data and struggles to generalise to fresh data. When applied to actual B2C settings, this can result in inaccurate predictions and decreased performance. Thus, enterprises with limited data (e.g. due to scale or low traffic) using B2C software may be unfit for DBN usage.

On the other hand, the proposed DBN has a variety of potential uses for e-commerce software quality. Some of the most important benefits include:

- Quality Prediction: Using targeted values for the KPIs, the DBN can be utilised to forecast the quality sub-characteristic values of e-commerce software. Before the software is made available to the general public, this can assist businesses in identifying potential quality concerns before they develop and take the appropriate action.
- Quality Optimisation: It is possible to optimise the quality of e-commerce software by changing the weights in the DBN. As a result, firms can more effectively manage resources by determining the most crucial KPIs and quality

sub-characteristics.

- Quality Control: The DBN can be used to keep track of the e-commerce software's quality as it is being developed. Organisations can see potential quality issues and take corrective action before the software is made available to the general public by monitoring the values of the KPIs and quality sub-characteristics over time.
- Quality Control: The DBN can be used to ensure that e-commerce software complies with the criteria for quality. Organisations can identify potential quality issues and take corrective action before making the software available to the general public by comparing the projected quality sub-characteristic values to the desired values.

The proposed DBN can potentially improve the quality of e-commerce software by providing organisations with a powerful tool for predicting, optimising, monitoring, and ensuring quality throughout the development process.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Rita, P., Oliveira, T. and Farisa, A. (2019) The Impact of E-Service Quality and Customer Satisfaction on Customer Behaviour in Online Shopping. *Heliyon*, **5**, e02690. https://doi.org/10.1016/j.heliyon.2019

[2] Pizzini, A., Vieira, R.B., Gomes, R.D., Santos, G., Malucelli, A. and Reinehr, S. (2021) Software Quality Practices in Growing Startups: A Qualitative Study. *Proceedings of the XX Brazilian Symposium on Software Quality* (*SBQS* '21), New York, 8-11 November 2021, 1-10. https://doi.org/10.1145/3493244.3493254

[3] Nistala, P., Nori, K.V. and Reddy, R. (2019) Software Quality Models: A Systematic Mapping Study. *Proceedings of the International Conference on Software and System Processes* (*ICSSP* '19), Montreal, 25 May 2019, 125-134. https://doi.org/10.1109/ICSSP.2019.00025

[4] Schnappinger, M., Osman, M.H., Pretschner, A., Pizka, M., and Fietzke, A. (2018) Software Quality Assessment in Practice: A Hypothesis-Driven Framework. *Proceedings of the* 12*th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (*ESEM* '18), Oulu, 11-12 October 2018, 1-6. https://doi.org/10.1145/3239235.3268922

[5] Aljarallah S. and Lock, R. (2020) A Comparison of Software Quality Characteristics and Software Sustainability Characteristics. *Proceedings of the* 2019 3*rd International Symposium on Computer Science and Intelligent Control* (*ISCSIC* 2019), New York, 25-27 September 2019, 1-11. https://doi.org/10.1145/3386164.3389078

[6] Stefani, A. (2020) A Metrics Ecosystem for Designing Quality E-Commerce Systems. *International Journal of Computer Science & Information Technology*, **10**, 1-15. https://doi.org/10.2139/ssrn.3615445

[7] International Standardization Organisation (2014) ISO 25000: Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Guide to SQuaRE.

https://iso25000.com/index.php/en/iso-25000-standards

[8] Stefani, A. and Xenos, M.K. (2011) Weight-Modeling of B2C System Quality. *Computer Standards & Interfaces*, **33**, 411-421. https://doi.org/10.1016/j.csi.2011.01.002

[9] Stefani, A. and Xenos, M.K. (2008) E-Commerce System Quality Assessment Using a Model Based on ISO 9126 And Belief Networks. *Software Quality Journal*, **16**, 107-129. https://doi.org/10.1007/s11219-007-9032-5

[10] International Standardization Organisation (2018) ISO 20000-1: Information Technology—Service Management—Part 1: Service Management System Requirements. https://www.iso.org/standard/70636.html

[11] Santos, D.S., Oliveira, B.R.N., Kazman, R. and Nakagawa E.Y. (2023) Evaluation of Systems-of-Systems Software Architectures: State of the Art and Future Perspectives. *ACM Computing Surveys*, **55**, Article No. 67. https://doi.org/10.1145/3519020

[12] Ndukwe, I., Licorish, S, Tahir, A. and MacDonell, S. (2023) How Have Views on Software Quality Differed over Time? Research and Practice Viewpoints. *Journal of Systems and Software*, **195**, Article ID: 111524. https://doi.org/10.1016/j.jss.2022.111524