Chapter 2
Theoretical and Conceptual Foundation

This chapter outlines the theoretical and conceptual foundation of the presented study. The investigation starts with a definition of the core concepts in Sect. 2.1. Drawing upon representational view of IT, which serves as central theoretical lens for this study, a novel definition of software sourcing modes is presented. IT alignment and its relationship with representational view of IT is discussed. Section 2.2 continues with a systematic literature review on software sourcing value. Software sourcing value is classified into an intermediate outcome in terms of software alignment and dependent concepts in terms of business process and sourcing performance. Section 2.3 gives a summary of the theoretical and conceptual foundation which serves as basis for the development of a preliminary research model in Chap. 3. The chapter design is given in Fig. 2.1.

2.1 Definition of Core Concepts

The presented study builds on the representational view of IT (Wand and Weber 1990). This section starts with a discussion of this theoretical perspective. Representational view of IT is applied to define the core concepts under study. In particular, a novel interpretation of software sourcing modes and the related in-house, on-premises, and on-demand settings is given. This section continues with a definition of IT alignment in general and shows its relationship to representational view of IT. A deeper discussion of the alignment between software and business processes is given in Sect. 2.2.

M. Nöhren, *Enterprise Software Sourcing Performance*, Progress in IS,
DOI 10.1007/978-3-319-23926-2_2

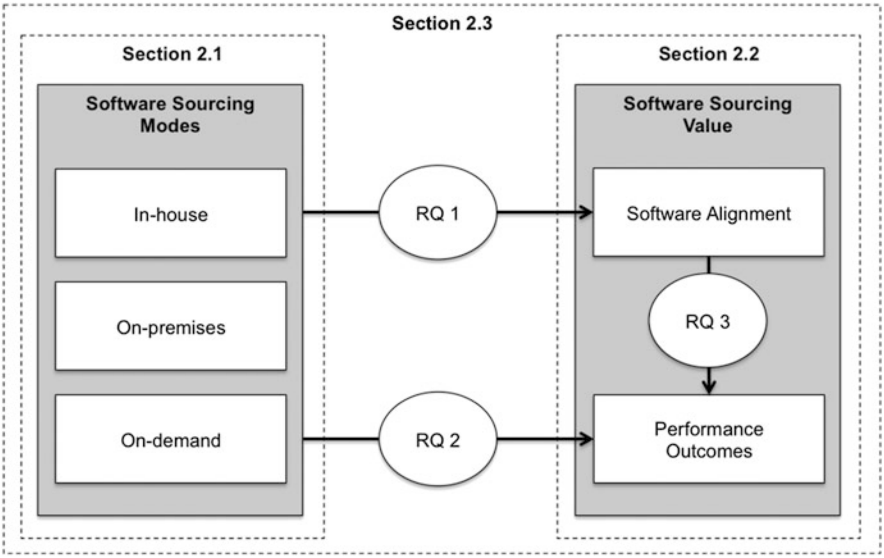

Fig. 2.1 Chapter design

2.1.1 Representational View of Information Technology

The representational view of IT (RIT) draws upon system's ontology. Ontology is a philosophical domain dealing with models of reality in terms of assumptions about how the world is made up and what the nature of things is (Guba and Lincoln 1994; Soh and Sia 2004). It defines how to describe the structure of the world in general (Wand and Weber 2002).

In information systems research, the Bunge-Wand-Weber (BWW) ontology has been applied to define the IT artefact (Soh and Sia 2004; Strong and Volkoff 2010). Instead of focusing on the way IT is managed, used, and implemented in organizations, and how these factors impact quality, performance, and adoption, BWW ontology views "*information systems as independent artefacts that bear certain relationships to the real-world system they are intended to model*" (Wand and Weber 1990: 61). In this view, information systems are seen as a representation of an organization and its reality (Wand and Weber 1990). Thereby, BWW seeks to understand what constitutes proper information systems by focusing on the properties an IT artefact needs to possess in order to fit with the requirements of a firm (Wand and Weber 1990). It is distinguished between physical, deep and surface structure elements of IT systems (see Fig. 2.2) (Wand and Weber 1990):

- **Surface structure** refers to the frontend of an IT system (Sia and Soh 2002). It is seen as a gateway where users interact with a particular IT artefact. Consequently, it describes how IT systems appear to their users (Wand and Weber 1995). Surface structure elements "*manifest the nature of the interface between*

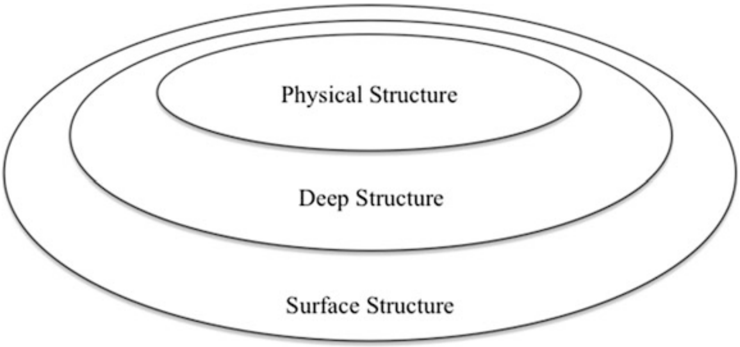

Fig. 2.2 Representational view on IT artefacts

the information system and its users and organizational environment." (Wand and Weber 1990: 61). It describes how real-world meanings are delivered by the system (Sia and Soh 2007).

- **Deep structure** refers to the core of the real-world system that an IT artefact is designed to model (Sia and Soh 2007; Strong and Volkoff 2010). It "*manifests the meaning of the real-world system the information system is intended to model*" (Wand and Weber 1995: 206). An IT artefact is made up of things that can either be real or conceptual in nature (Sia and Soh 2007). These things have intrinsic properties attached and exist at a certain state, which is changed through transformation (Sia and Soh 2002, 2007).
- **Physical structure** refers to the technology used to implement an IT system (Wand and Weber 1990, 1995). It explains "*ways in which deep and surface structures are mapped onto underlying physical technology*" (Strong and Volkoff 2010: 750). Examples of physical structures are networks, printers, or mass-storage devices (Strong and Volkoff 2010; Wand and Weber 1990).

Drawing upon the distinction between physical, deep, and surface structure elements, the software artefact was recently defined from a RIT perspective (Sia and Soh 2007; Strong and Volkoff 2010). This definition of software serves as a basis for a novel theory-guided interpretation of in-house, on-premises, and on-demand sourcing modes, which is presented afterwards.

2.1.2 Software Sourcing Modes

Software sourcing encompasses two distinct components that needs to be defined. From a RIT perspective, **software** is seen as a set of deep and surface structures mapped onto physical IT infrastructures (Sia and Soh 2007; Strong and Volkoff 2010). The structural elements of software artefacts are given in Fig. 2.3 (Sia and Soh 2007).

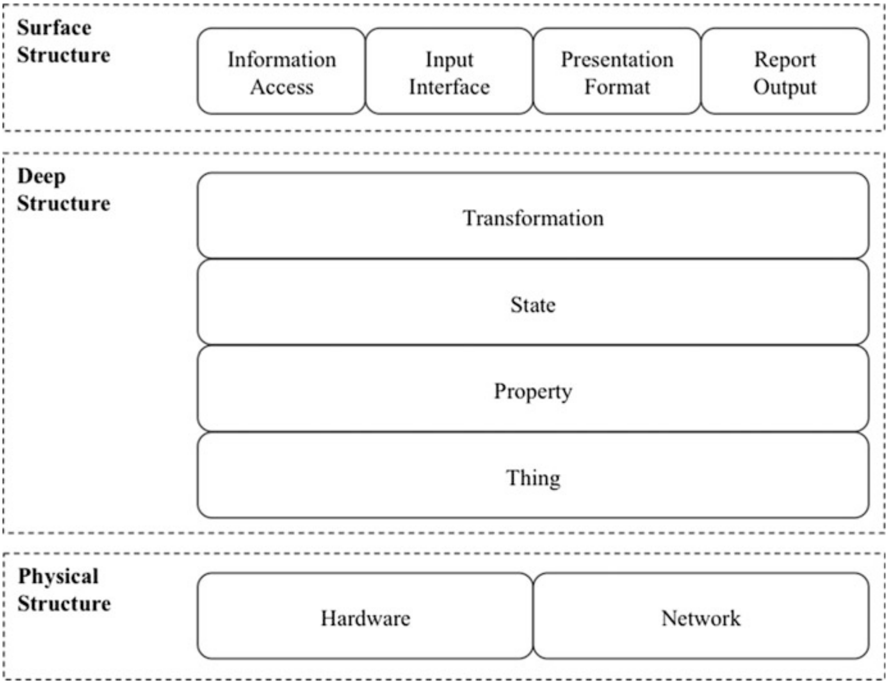

Fig. 2.3 Representational view on software artefacts

Starting with the deep structure, software artefacts are made of *things* (Sia and Soh 2007). Such things are the most elementary units in RIT (Sia and Soh 2007; Wand and Weber 1990). If a thing is connected with other things, a system is formed (Wand and Weber 1990). Inventory items, customer orders, or supplier accounts are examples for things related to software artefacts (Sia and Soh 2007).

All things must have certain *properties* attached (Sia and Soh 2007). These properties are functions that map things into value (Sia and Soh 2007; Wand and Weber 1990, 1995). Examples of software properties are inventory numbers, sales amounts, or unit prices stored within the system (Sia and Soh 2007).

The *state* of a thing refers to the vector of all property values of the thing (Wand and Weber 1990). Accordingly, a state is a set of conditions a thing might take, such as a status of a production order (Sia and Soh 2007).

A *transformation* of a thing is a change in its state (Wand and Weber 1990). Each transformation has a certain space—a set of possible changes that can occur in a thing—and a transformation law—rules that define which changes are legal—attached (Sia and Soh 2007; Wand and Weber 1990). Examples for software transformations are rules for production planning and execution, or the calculation of cost (Sia and Soh 2007).

While deep structures describe the core of a software artefact, surface structure refers to the user interface of an application (Sia and Soh 2007; Wand and Weber

1995). Each software artefact possesses a certain *input interface*, a particular *data representation format*, specific rules for *information access*, and a defined *report output* (Sia and Soh 2007). Examples of software surface structures are input parameters, content of production reports, and formats of order documents (Sia and Soh 2007).

Physical structures refer to the underlying technology linked to the software artefact (Strong and Volkoff 2010; Wand and Weber 1990). Despite the fact that these structures are not directly part of a particular software artefact, they are essential to implement, operate, and use an application (Strong and Volkoff 2010). Physical structures refer to the *hardware* on which an application is installed and to the *network* infrastructures that link systems and users with each other.

Having specified the software artefact, this section continues with a definition of software sourcing taking a RIT perspective. In general, **sourcing** refers to the procurement of goods and services from internal and external entities. It can be defined as turning over parts of a corporate's IT function to a third party vendor, who in exchange provides and manage IT assets and services for monetary returns over an agreed period of time (Apte et al. 1997; Kern 1997). Such a corporate's IT function includes three subfunctions that can be outsourced (Dibbern and Heinzl 2009; Heinzl 1993):

- **System development function** (e.g. development of own applications, adaptation of standard software)
- **System operation function** (e.g. maintenance of existing applications, implementation of updates)
- **Management function** (e.g. planning, coordination, and controlling of systems)

Organizational acquisition arrangements for these three subfunctions can be categorized according to the two dimensions *ownership* and *location* (Murray et al. 2009; Tanriverdi et al. 2007). The first one refers to the degree of vertical IT resource integration (Anderson and Parker 2002). It is differentiated between insourcing (hierarchical governance) as well as pure and hybrid outsourcing approaches (market governance) such as selective outsourcing, bi-sourcing, joint ventures, or strategic alliance sourcing (e.g. Currie and Willcocks 1998; Du et al. 2009; Kim and Park 2010; Lacity and Willcocks 1998; Lee et al. 2004). Insourcing defines an acquisition strategy where IT resources and all related operational and management activities are held within an organization's hierarchy (e.g. Cha et al. 2009; Kern et al. 2002a, b; Kishore et al. 2004). Contrary to this, outsourcing refers to an acquisition strategy where IT resources, IT tasks, and IT activities are contracted out to an external provider (e.g. Kern et al. 2002a, b; Kern 1997).

Location refers to the global position of IT subfunctions (Heinzl 1993). It can be differentiated between onshoring, nearshoring, and offshoring of system development, system operation, and management (Davis et al. 2006; Heinzl 1993; Nöhren et al. 2013). The global location of IT subfunctions is significant in understanding performance of globally executed software development and maintenance projects (e.g. Dibbern et al. 2008; Schwarz et al. 2009) and in analysing the role of global

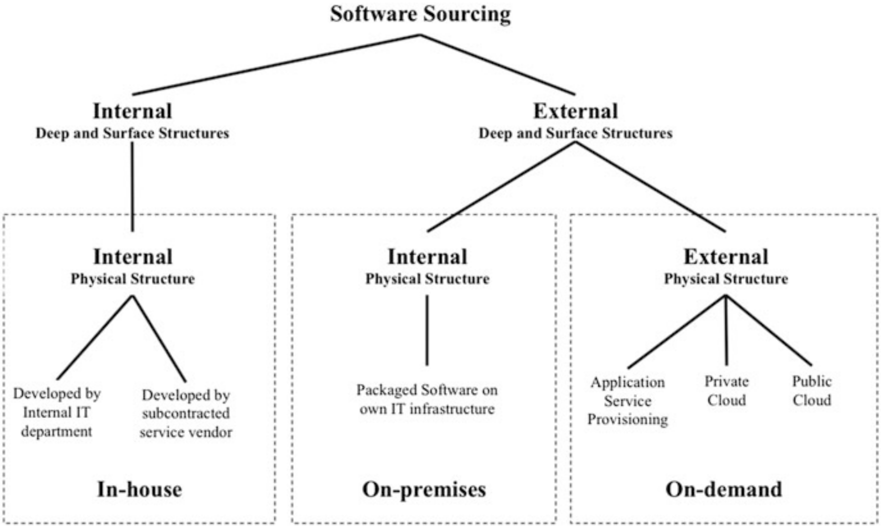

Fig. 2.4 Representational view on software sourcing modes

network structures in delivering outsourced tasks and activities (e.g. Ang and Inkpen 2008; Nöhren and Heinzl 2012). In this study, software sourcing modes are defined taking an ownership perspective. The global location of software vendors is not further investigated.

Looking at literature in the area of software sourcing, no strong definition of the phenomenon exists (e.g. Banker and Kemerer 1992; Choudhary 2007a; Kern et al. 2002a, b; Schwarz et al. 2009; Walden 2005; Wang 2002). However, "*(...) as software becomes an ever-increasing proportion of information technology, it is necessary to update IT outsourcing theory to consider the different types of ownership that software allows*" (Walden 2005: 699). Drawing on RIT, different software sourcing arrangements are classified according to their ownership on physical, deep, and surface structures and are categorized into in-house, on-premises, and on-demand modes (see Fig. 2.4).

The in-house mode encompasses applications where physical, deep, and surface structures of the software artefact are held within a firm's hierarchy. In this sourcing arrangement, an application is custom-developed for a specific firm (Schwarz et al. 2009). The development is either performed by an internal IT department or a subcontracted software vendor under control of the client (e.g. Nidumolu 1995; Schwarz et al. 2009; Wang 2002). The enterprise system is installed on a corporate's IT infrastructure representing an internal deployment. In contrast to this, the ownership on deep and surface structures of packaged applications is held by a software vendor. The design of these structures is impacted by requirements of a large customer base resulting in inherent software structures determined by industry's best practices (Maurer et al. 2012; Sia and Soh 2007; Strong and Volkoff 2010). Packaged systems differ with respect to their physical structures in terms of their deployment. They can be classified into a subscription-based on-demand model and perceptual-licensing for on-premises applications (Choudhary 2007a; Winkler et al. 2011). Whereas on-premises applications are installed on a firm's own IT infrastructure (internal physical structures), on-demand applications are hosted with a software vendor (external physical structures) and are accessed via Internet (Winkler et al. 2011). As a result, we see a decreasing extent of ownership on physical, deep, and surface structures from in-house through on-premises to on-demand sourcing.

2.1.3 Information Technology Alignment

As outlined above, RIT sees the IT and software artefact as a representation of a real-world system in terms of an organization (Strong and Volkoff 2010; Wand and Weber 1990). If real-world things, properties, states, and transformations (deep structure) are insufficiently represented by the artefact, or if a system's interface (surface structure) differs from the way people access, input, or retrieve information in reality, misalignments occur (Sia and Soh 2007). In this perspective, the degree of alignment can be seen as the extent of IT representing organizational reality. The presented study is rooted into previous research on IT alignment. A deeper definition of software alignment from a RIT perspective is given in the next section.

Alignment has been conceptualized and defined in various ways. It can be viewed as a "*(...) state in which the goals and activities of a business are in harmony with the information systems that support them*" (McKeen and Smith 2006: 93). For more than two decades, alignment of IT and business has been among the top five management concerns (Chan and Reich 2007; Luftman and Ben-Zvi 2010; Luftman et al. 2013). IT and business leaders see alignment as a key enabler of efficiency and effectiveness and therefore focus on initiatives that enhance the maturity of alignment between IT and business (Luftman et al. 2013). Against this background, numerous studies have been conducted, investigating positive outcomes of alignment such as an increased competitive advantage (e.g. Floyd and Woolridge 1990; Kearns and Lederer 2003) as well as organizational and business process performance (e.g. Bergeron et al. 2004; Byrd et al. 2006; Choe 2003; Tallon 2007). Other studies focused on antecedences of alignment such as shared decision making between IT and business units (Kearns and Lederer 2003) as well as the role of these units' shared understanding (Preston and Karahanna 2008; Reich and Benbasat 2000). Furthermore, three alignment cultures in organizations were identified (Ravishankar et al. 2011) and how alignment and related capabilities evolve over time (Sabherwal et al. 2001) was studied.

Alignment research varies with respect to the level of analysis. Chan and Reich (2007) distinguish between studies on organizational level (e.g. Kearns and Lederer 2003), system and project level (Floyd and Woolridge 1990; Grant 2003; Tallon 2007), as well as individual level (Tan and Gallupe 2006) such as the task-technology-fit model (Goodhue and Thompson 1995). The presented study defines software sourcing from an outsourcing perspective. In particular, software sourcing is seen as outsourcing of parts of a corporate's IT function on a system's level.

According to a literature review by Chan and Reich (2007), prior research on IT alignment can be classified into five dimensions. First, on an **informal structure dimension**, the distinct nature between informal and formal organizational set-ups were compared (Chan and Reich 2007). It had been discovered that the informal structure serves as a key antecedence for achieving IT alignment. For instance, Chan (2002) found that a fit between IT and business strategies had resulted in higher levels of performance. It was claimed that this was rather a result of informal organizational structures in terms of trust, commitment, or friendship between business and IT managers than a consequence of formal and instituted organizational procedures (Chan 2002).

The second dimension focuses on **cultural alignment** by studying the cultural fit between IT and business (e.g. Chan and Reich 2007; Ravishankar et al. 2011). For instance, by focusing on the implementation of a knowledge management system within a case company, Ravishankar et al. (2011) investigated the impact of organizational cultures and subcultures on realizing strategic IT alignment. Two important findings were derived. First, three organizational cultures—referred to enhancing, countercultural, and chameleon subculture—shaping the effects of top-down versus bottom-up system implementation were identified. Second, it was suggested, that the same system could result in alignment in one organizational unit whereas it leads to misalignment in another department (Ravishankar et al. 2011).

Third, on a **social dimension**, IT alignment refers to "the state in which business and IT executives understand and are committed to the business and IT mission, objectives, and plans" (Reich and Benbasat 2000: 81). Studies in this field investigated the role of shared understanding, shared knowledge, and communication processes among people involved in IT planning (e.g. Preston and Karahanna 2008; Reich and Benbasat 2000). Research focused frequently on the formal relationship between business and IT managers. To give an example, by taking a resource-based view of the firm perspective (see Sect. 2.2.1), Kearns and Lederer (2003) studied how social exchange between CEO and CIO resulted in higher levels of strategic IT alignment. In particular, the effect of reciprocal structures, where CEOs participate in IT planning and CIOs participate in business planning was studied. It turned out that this social alignment resulted in significant higher levels of strategic alignment (Kearns and Lederer 2003).

Particular attention had been paid to structural and strategic alignment (Chan and Reich 2007). Previous studies found a positive impact of both dimensions on process as well as organizational performance (Bergeron et al. 2001; Powell 1992; Tallon 2007). **Strategic alignment** describes the extent to which a certain business plan is in line with a corporate's IT plan and how IT strategy supports business strategy (Chan and Reich 2007; Henderson and Venkatraman 1993; Kearns and Lederer 2003). **Structural alignment** refers to the fit between structures and processes of a business unit and those of the IT system (e.g. Aier and Winter 2009; Chan 2002; Henderson and Venkatraman 1993). The presented study combines the strategic and structural dimension of IT alignment. As it will be shown in this work, dynamic fit, which serves as measure of software alignment, is related to the structural dimension of IT alignment (Maurer et al. 2012; Nöhren et al. 2014; Strong and Volkoff 2010). The process by which dynamic fit is generated is transferred into strategic alignment pattern referred to as gestalt and non-gestalts in Chap. 3 (Mintzberg 1987; Sabherwal and Robey 1995; Venkatraman 1989). These alignment patterns are related to the strategic dimension of IT alignment.

2.2 Literature Review

Having defined software sourcing modes and IT alignment from a RIT perspective, this section continues with a systematic literature review in order to define software sourcing value and concepts related to performance outcomes and software alignment. The analysis is organized as follows: Sect. 2.2.1 starts with a definition of additional theoretical lenses that were identified by the literature review. These reference theories were found to be valuable in defining the dynamic fit process (Sect. 2.2.5) and in building a preliminary research model in Chap. 3. Section 2.2.2 defines IT value in general and presents previous process models on how IT artefacts contribute to performance of organizations and business processes. These process models are synthesized into a more generic model on IT value generation that outlines the relationship between software alignment and performance outcomes. Section 2.2.3 continues with a definition of performance outcomes. Drawing upon the literature review, two dependent concepts in terms of business process and sourcing performance have been identified. Earlier contribution to software alignment is discussed in Sect. 2.2.4. The concept of dynamic fit is developed based on three previously published notions of software fit. Due to the fact that the developed dynamic fit concept is non-deterministic and dynamic in nature, the process by which dynamic fit is generated and related concepts are outlined in Sect. 2.2.5.

The literature review followed the guidelines by Kitchenham et al. (2009) as well as Webster and Watson (2002). After the research questions were clarified in the first chapter of this study, the search process including a definition of keywords and relevant sources of knowledge was designed. As shown in Appendix A, a combination of two types of keywords was used. Whereas the first type focused on the IT artefact under study, the second row included keywords related to performance outcomes. As indicated by Webster and Watson (2002), major contribution to a specific research topic is most likely to be found in highest ranked publication outlets of a discipline. Therefore, this literature review focused on the top eight IS journals as defined by the Association for Information System's senior scholars' basket of journals.[1] In addition, the leading German-based international journal on

[1] http://ais.site-ym.com/?page=SeniorScholarBasket

management information systems as well as the top five international conferences as shown in Appendix A were included in this review.

The usage of very general keywords led to more than 8000 search results. Consequently, a second search, in which keywords were limited to abstracts only was conducted. Thereby, the total number of search results was reduced to less than 400 papers. Those papers' abstracts, introductions, discussions, and conclusions were read (Dibbern et al. 2004) in order to determine whether the study was helpful in answering the research questions (Kitchenham et al. 2009). Remaining papers had been combined with a forward and backward search as recommended by Webster and Watson (2002).

2.2.1 Theoretical Lenses in Software and Sourcing Research

Having outlined RIT as central theoretical lens of this investigation, this section continues with a definition of additional reference theories identified by the literature review. Looking at research on software sourcing, a variety of reference theories were applied to explain performance outcomes. An overview of those theoretical lenses as well as their respective elucidations of software sourcing value is given in Table 2.1.

2.2.1.1 Organizational Structure Theories

Institutional theory and RIT frequently served as reference theories for studying the fit between organizations and its IT systems (e.g. Chiasson and Green 2007; Nöhren et al. 2014; Sia and Soh 2007; Soh and Sia 2004; Strong and Volkoff 2010). These theoretical lenses are particularly helpful in defining structures related to the organization and those related to a certain IT artefact under investigation as well as in understanding external and internal contingencies shaping those structures (Scott 1987; Sia and Soh 2007; Strong and Volkoff 2010; Wand and Weber 1990).

Institutional theory (Scott 1987) is related to the nature of organizational structures. The core concept of this theory is the institution, which is seen as an organized and established procedure (Berente and Yoo 2012). Institutional theory attends to the deeper and more robust aspects of social structures. In particular, institutional theory focuses on how structures, norms, and rules are shaped by a social and an organizational context and how they become established over time (Berente and Yoo 2012; Jones and Karsten 2008). This context is primarily driven by an organization's internal and external environment and a firm's interaction with it (Lu and Ramamurthy 2010; Sia and Soh 2007). Institutional theory argues that longevity and survival of an organization can only be achieved when a firm remains consistent with changing internal and external environmental conditions over time (Scott 1987; Vessey and Ward 2013).

Table 2.1 Theoretical lenses on software sourcing

Paradigm	Reference theory	Key contribution to software sourcing	References in software sourcing literature
Organizational structure theories	Representational view of IT (RIT)	Identification of inherent characteristics of software resulting from deep and surface structures; explanation of software-business process fit	Nöhren et al. (2014), Sia and Soh (2007), Strong and Volkoff (2010)
	Institutional theory	Role of external and internal contingencies defining environmental, business process and software structure change	Sia and Soh (2007), Soh and Sia (2004), Xin and Levina (2008)
Organizational resource theories	Resource-based view of the firm (RBV)	Identification of inherent characteristics of business processes	Benlian et al. (2009), Schwarz et al. (2009)
	Resource-dependence theory (RDT)	Identification of relational characteristics in software sourcing relationships	Schwarz et al. (2009)
	Knowledge-based view of the firm (KBV)	Identification of inherent characteristics of business processes	Schwarz et al. (2009), Winkler and Brown (2014)
Organizational governance theories	Transaction cost economics (TCE)	Identification of inherent characteristics of software artefacts	E.g. Schwarz et al. (2009), Susarla et al. (2009), Winkler and Benlian (2012), Winkler and Brown (2014)
	Agency theory	Understanding risks and costs in sourcing settings; particularly relevant for customized software development	Winkler and Brown (2014)

2.2.1.2 Organizational Resource Theories

Organizational resource theories, including resource-based view of the firm, resource-dependence theory, and knowledge-based view of the firm see companies as collection of resources, which are central to firm strategies (Dibbern et al. 2004; Winkler 2009). **Resource-based view of the firm** (RBV) is a widely acknowledged theoretical lens on describing, explaining, and predicting organizational relationships (Barney et al. 2011). RBV defines a corporate resource as "*an asset or input to production (tangible or intangible) that an organization owns, controls, or has access to*" (Helfat and Peteraf 2003: 999). Such human, physical, and organizational resources are mandatory to fulfil a company's task and result in a sustained competitive advantage if they are valuable, rare, and imperfectly mobile (Barney 1991; Dibbern et al. 2004; Mata et al. 1995; Nöhren and Heinzl 2012). A resource is valuable if it enables a firm to conceive or implement a strategy, which improves its efficiency and effectiveness (Barney 1991). It is seen as rare, if it is not possessed by a large number of competitors (Barney 1991). Imperfect mobility or non-tradability means that a resource cannot be imitated by competitors or substituted by any another resource (Barney 1991; Wade and Hulland 2004).

RBV had widely been used to study the value of IT (e.g. Bharadwaj 2000; Melville et al. 2004; Mithas et al. 2011). According to Mata et al. (1995) three attributes of IT can be a source of sustained competitive advantage. First, building superior IT assets is typically very cost intensive (see also Soh and Markus 1995). Corporate's *access to capital* serves as a source of competitive advantage. Second, a *proprietary technology* protected by patents or other security mechanisms can prevent a valuable IT resource from being copied (see also Teece 1986). Third, *managerial IT skills* in terms of management's ability to build and exploit IT applications that support business units are imperfectly mobile in nature. Such skills are developed over long periods of time. In contrast to this, *technical IT skills*, which refers to the know-how needed to execute IT-related tasks such as building, maintaining, or implementing IT applications, are rather a source of temporary competitive advantage (Mata et al. 1995). In a later study, Bharadwaj (2000) defined IT as a set of IT infrastructure resources, human IT resources, and intangible IT-enabled resources such as know-how and corporate culture. *Human resources* encompass technical and managerial IT skills. *IT infrastructure resources* are physical assets such as computers, communication technologies, platforms, and databases. Finally, *IT-enabled intangibles* are organizational resources like knowledge assets, synergies, and a firm's customer orientation (Bharadwaj 2000).

Whereas RBV focuses on internal resources, **resource dependency theory** (RDT) takes an outside perspective. In particular, RDT argues that organizations are not able to produce and provide all resources needed internally (Pfeffer and Salancik 1978; Schwarz et al. 2009). To varying degrees, all firms depend on some resources of their external environments and must actively engage in managing their ecosystems as well as their resource flow (Borman 2006; Dibbern et al. 2004; Pfeffer and Salancik 1978; Sia et al. 2008). By outsourcing parts of the IT function, firms increase their dependency on their vendors (Sia et al. 2008). This is particularly true in the context of sourcing packaged applications by outsourcing deep and surface structure of an IT artefact. To avoid a strategic vulnerability resulting from resource dependency, sourcing options need to be carefully evaluated (Kern et al. 2002a, b). Due to the fact that RDT rather focuses on how to manage an ongoing outsourcing relationship, how to increase power over a strategic resource from external environment, and explain why firms procure packaged applications instead of building their own systems (Schwarz et al. 2009), this theoretical lens is not further investigated in this study.

Finally, **knowledge-based view of the firm** (KBV), as third perspective on corporate resources, can be seen as a spin-off of RBV (Barney et al. 2011). It centres knowledge as the most significant resource of the firm and claims that, due to the fact that knowledge-intensive resources are difficult to imitate (imperfect mobility) and are heterogeneously distributed among firms (rare), such resources are key factors of sustained competitive advantage and performance (Dibbern et al. 2008; Grant 1996; Herath and Kishore 2009). Competitive advantage results from a firm's ability to create, store, and apply knowledge (Jayatilaka et al. 2003; Kishore et al. 2004). Rather than focusing on physical, organizational, and human resources, KBV emphasized on the knowledge required for development and deployment of enterprise systems (Jayatilaka et al. 2003). KBV is often seen as complementary to transaction cost economics (see below) in explaining why firms engage in activities, which generate superior and valuable internal knowledge but appears to be inefficient from an economic point of view (e.g. sourcing of services internally while a hazard-free external market exists) (Reitzig and Wagner 2010).

2.2.1.3 Organizational Governance Theories

Transaction cost economics (TCE) is a widely used theory of the firm. Developed by Williamson (1973, 1979) and drawing upon Coase (1937), it argues that firms exist because using markets would be too costly. Firms boundary decisions are framed as "make-or-buy"-problems (Reitzig and Wagner 2010). TCE analyses the relative advantage of using an internal (hierarchy) or external (market) governance mode by focusing on costs of transactions (Sia et al. 2008). The key assumption is that firms aim at minimizing transaction costs when selecting a particular governance mode. These costs depend on asset specificity, uncertainty, and transaction frequency of an activity (Sia et al. 2008). *Asset specificity* refers to the degree to which an application can be redeployed to another context (Wang 2002). It "*takes a variety of forms—physical assets, human assets, site specificity, dedicated assets, brand name capital, and temporal specificity—to which individuated governance structure responses accrue*" (Williamson 1998: 36). If *asset specificity* is high, contractual partners have to provide specific investments in an outsourcing process that have little or no value to them (Wang 2002). *Uncertainty* refers to the risk of unforeseen contingencies that—for instance—result in renegotiations or opportunistic behaviour of vendors (Sia et al. 2008; Wang 2002). Finally, *frequency* is related to how often a transaction occurs (Wang 2002). Activities that are highly asset specific, that involve a high level of uncertainty, and that occur regularly are rather performed by an internal governance mode than by outsourcing (Gilley et al. 2004). It was found that out of these three components, asset specificity had the most consistent explanatory power in a wide range of empirical studies (Dibbern et al. 2008).

Agency theory is closely linked to the uncertainty construct of TCE (Bahli and Rivard 2003). It describes a situation in which a principal (client) engages with an agent (vendor) to perform a service (Hustad and Olsen 2011). Uncertainty arises from asymmetric information between the contractual parties (Dibbern et al. 2004). Conflicting goals, incomplete information, and different risk perceptions between principals and agents demand for appropriate accountability schemes to generate the expected value from the relationship (Winkler and Brown 2014). Consequently, principals have to set initiatives (e.g. in terms of contracts) to ensure that the agents behave in the expected way and to reduce the risk of moral hazard (Dibbern et al. 2004; Gefen et al. 2008). Due to the fact that the key focus of agency theory is on an optimal contractual arrangement between principal and agent in order to reduce costs and risks (Dibbern et al. 2004) than on analysing performance in sourcing of IT artefacts, agency theory was not further investigated. Agency theory is seen as particularly useful in studying customized software development tasks on a project level (e.g. Gefen et al. 2008).

2.2.2 The Value of Information Technology

Following a brief description of reference theories identified by the literature review on software sourcing, this section continues with a deeper discussion of the IT value term. Value refers to "*the importance, worth, or usefulness of something*" (Oxford Dictionaries 2014c). Ever since Nobel Prize winner Robert Solows famous quote in 1987 "*we see the computer age everywhere except in the productivity statistics*" (Brynjolfsson and Hitt 1998: 51), a huge body of research emerged, concerned with analysing and measuring IT's value for businesses, processes, organizations, and economies (e.g. Bharadwaj et al. 1999; Brynjolfsson and Hitt 1998; Chan 2000). It was found that "*firms derive business value from IT through its impacts on intermediate business processes. Such intermediate processes include the range of operational processes that comprise a firm's value chain and the management processes of information processing, control, coordination and communication.*" (Mooney et al. 1996: 69). This "*IT value often lies in the cross-functional integration of business processes and the penetration of IT into the core of organizational functioning.*" (Willcocks et al. 2002: 51).

Despite a vast number of studies on the value of IT, the debate on how IT contributes to performance still persists (Melville et al. 2004). The link between IT and productivity of a firm has widely been discussed but remained fuzzy and is little understood (Brynjolfsson 1993). Previous research can be classified with respect to three dimensions based on the questions of "where", "what", and "how" revolving around IT value. One of the major discussions revolves around the question of "where" to analyse and find the value of IT. Previous studies investigated IT outcomes on different organizational levels. With respect to this **level of analysis**, IT value research can be classified into three streams. The first stream is concerned with measuring value on a corporate level. Studies on this level analysed the impact of aggregated IT spendings, internally provided IT and non-IT capital, as well as IT strategy on market share and return on assets (Barua et al. 1995; Bharadwaj et al. 1999; Dehning and Richardson 2002). Most of these studies are econometrical in nature. A second stream of research focuses on individual level impacts. Analysis in this field emphasized the importance of IT design and IT management on users behaviour as critical factors for achieving desired organizational goals (e.g. Au et al. 2008; DeLone and McLean 1992, 2003; van der Heijden 2004a; Soh and Markus 1995). These studies were frequently linked to research in the field of social psychology. Finally, a third stream of research is on a process level assessing IT's value within corporate's IT functions and business units. It stresses the significance of IT resources in realizing and improving business process performance (Dohmen et al. 2010; Melville et al. 2004; Tallon et al. 2000).

The second dimension refers to the **IT anchoring** and the question of "how" IT influences outcomes. With respect to the relationship between predictor and dependent variable, previous research can be classified into criterion-free and criterion-specific definitions of IT value. The first one describes the IT artefact or its components as antecedence for achieving performance outcomes. In this perspective, IT drives efficiency and effectiveness of firms, processes, and individuals. For instance, Bharadwaj (2000) analysed the impact of superior IT capabilities that encompassed human IT resources, IT infrastructure resources, and IT-enabled intangibles on firm performance. Melville et al. (2004) found that human and technological IT resources enable business processes and impact their performance. The criterion-specific perspective defines IT as input to production. Performance is seen as a transformation of these inputs into desired organizational outputs. The IT artefact, the IT function, or its components are parts of the performance definition. For instance, Dehning and Richardson (2002) stressed how IT-related inputs such as IT spending and IT capital are related to certain levels of outputs. Barua et al. (1995) analysed how factors such as IT and non-IT capital as well as IT and non-IT purchases impact process (e.g. new product development and inventory turnover) and organizational performance outcomes (e.g. market share and return on assets). Banker et al. (1990) conducted a field experiment analysing performance gains of subsidiaries that used a particular business application in comparison to those where the application was not deployed.

Finally, the third question is concerned with "what" **IT impact** actually is. A huge body of research focused on the outcomes of IT in general as well as the performance of specific systems. Thereby, the notion of outcome is used in a vague and inconsistent way (e.g. DeLone and McLean 1992; Dibbern et al. 2004; Schryen 2010). Some studies defined outcome in terms of performance assessing the output from an economic point of view by including measures such as market value, operational efficiency, productivity, and profitability of IT (e.g. Brynjolfsson and Hitt 2000; Dehning and Richardson 2002; Heinrich et al. 2011; Kohli and Devaraj 2003; Melville et al. 2004). Other studies relied on the notion of success which describes the effectiveness of a system in terms of its actual and perceived contribution to operational, strategic, economic, or technological benefits (e.g. DeLone and McLean 1992, 2003; Grover et al. 1996; Lee et al. 2004). A deeper discussion of the dependent variable is given in Sect. 2.3. The position of this study is shaded black in Fig. 2.5.

Having classified the three dimensions of IT value research, this section continues with a discussion of four previously published process models on IT value. These models are synthesised into a more generic model which reveals the relationship between alignment and performance outcomes.

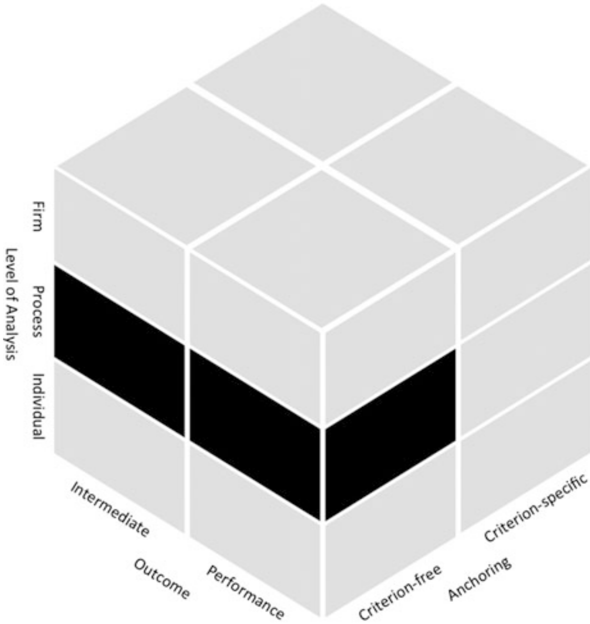

Fig. 2.5 Three dimensions of IT value research

2.2.2.1 The Soh-Markus-Model on IT Investments

One of the earliest models on returns from IT spending is the business value model by Soh and Markus (1995). Based on a review of five previously proposed IT outcome models, the authors introduced a novel process model to capture the value of IT. This model relies on the criterion-specific definition of outcomes. Soh and Markus (1995) argued that *IT expenditures* lead to *organizational performance* in three consecutive processes (see Fig. 2.6). Within the first process, referred to as "IT conversion process", *IT expenditures* are seen as necessary conditions to build *IT asset*. *IT assets* in terms of superior applications, infrastructures, skills, or knowledge arise when IT spending is transformed in an efficient and effective way. IT management such as project management and the formulation of an adequate IT strategy impacts this conversion process. Therefore, the "IT conversion process" takes place within the IT function of an organization (Soh and Markus 1995).

Following the first process, the "IT use process" within the business units begins. If a company had been able to transform *IT expenditures* into *IT assets*, these assets form the basis to derive certain *IT impacts*. These impacts in terms of new product developments, redesigned work pattern, or improved decision-making, can be found on an organizational and on a business process level. However, like *IT expenditures*, *IT assets* are merely necessary conditions to realize *IT impacts*. Within the second process, actual and appropriate usage of the IT system by its key stakeholder within the business units was defined as condition to realize desirable impacts. Finally, the third process, following the "IT use process"

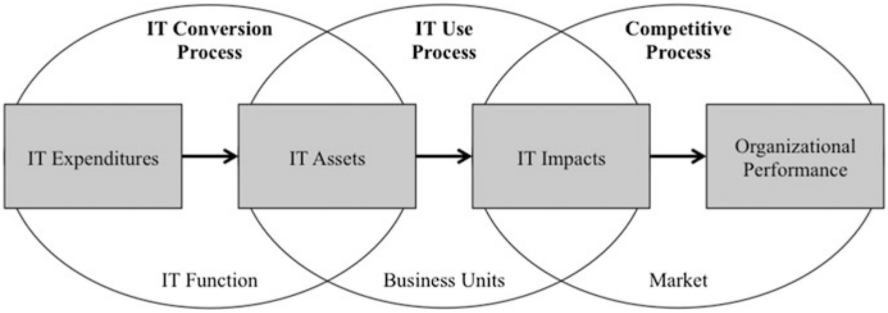

Fig. 2.6 The Soh-Markus-Model on IT investments based on Soh and Markus (1995)

describes the transformation of *IT impacts* into *organizational performance*. Thereby, it is differentiated between the notion of *IT impacts* referred to as value directly linked to a particular IT artefact and *organizational performance* as the indirect outcome of *IT assets*. *IT impacts* were described as necessary condition for *organizational performance* in terms of firm productivity or financial and stakeholder value. *Organizational performance* is realized on a market and is therefore mainly impacted by environmental factors such as the competitive position of a firm or its competitive dynamics. Due to the fact that the IT value generation process starts with certain IT spending, the Soh-Markus-model can be referred to as an investment-based IT value model (Soh and Markus 1995).

2.2.2.2 The Dehning-Richardson-Model on IT Investments

A second investment-based IT value model was introduced by Dehning and Richardson (2002). A systematic literature review was conducted and previous findings for the criterion-specific dimension of IT value were synthesized. It was found that foregoing research conceptualized IT-related investments with respect to (1) *strategy*, (2) *management*, and (3) *financial spending*. With respect to *IT strategy*, it was found that earlier studies assessed the impact of different types of IT deployment (e.g. ERP systems), time of IT sourcing (e.g. first-mover advantages), or the facilitation of new business strategies. Researchers investigating *IT management* emphasized the role IT and related capabilities hold within an organization. Finally, studies on *IT spending* measured investments in terms of total IT expenditures, costs of IT training, and staff costs (Dehning and Richardson 2002). By including IT strategy, IT management, and IT spending in the definition of IT inputs, the pure focus on financial expenditures like in the Soh-Markus-model (Soh and Markus 1995) was extended (see Fig. 2.7).

Key focus of this study was on exploring opportunities to capture returns of IT. A conceptual framework for measuring outcomes of IT investments on different levels was introduced (see Fig. 2.6). In particular, Dehning and Richardson (2002) derived a set of variables to capture outcomes of *IT spending*, *IT management*, and

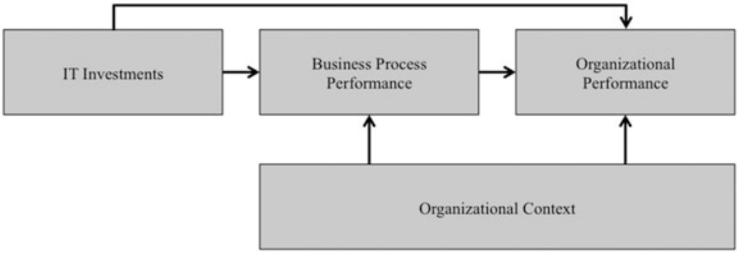

Fig. 2.7 The Dehning-Richardson-Model based on Dehning and Richardson (2002)

IT strategy on business process and organizational level. Process level concepts were related to measures such as inventory turnover, quality, and efficiency. It was found that IT investments result in direct and indirect effects on business processes performance. Direct effects referred to impacts such as reduction of inventory costs as output of transforming *IT spending*, *IT management*, and *IT strategy* inputs. Indirect effects are "collateral benefits" of IT investments, such as an improved quality of managers' decision-making following an ERP investment, which in return results in new opportunities for business processes. On an organizational level, Dehning and Richardson (2002) identified several market (e.g. event study, market value) and accounting measures (e.g. return on assets, market share). In addition, the crucial role of organizational contingencies such as industry sector, firm size, or competition shaping business process and organizational performance was exposed (Dehning and Richardson 2002).

2.2.2.3 The DeLone-McLean-Model on IT Systems

One of the most famous and most widely cited models in IT value research is the so-called DeLone-McLean-model of IT success (see Fig. 2.8). In this model, IT is seen as antecedence for achieving desirable outcomes. Drawing upon a review of 180 articles in the area of IT success, a process model to capture the complex nature of IT impacts on a system's level was developed. In particular, two central challenges in assessing IT outcomes were faced. The first one was the search of the dependent variable. Six IT value concepts were identified and included within the model. The second one was the question of how these concepts are interrelated with each other. A process model as shown in Fig. 2.8 was derived (DeLone and McLean 1992).

DeLone and McLean (1992) pointed out that due to the fact that IT value is typically defined in an elusive and vague way, almost every study relied on its own measures. In order to provide a more comprehensive picture, six identified IT outcome concepts in terms of *system quality*, *information quality*, *use*, *user satisfaction*, *individual impact*, and *organizational impact* were included within the model. The two concepts of *system quality* and *information quality* were directly attributed to the IT artefact and served as measures for *IT system performance*.

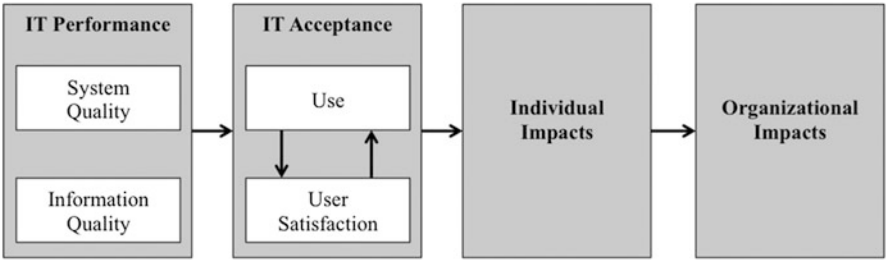

Fig. 2.8 The DeLone-McLean-Model based on DeLone and McLean (1992)

System quality describes the technical dimension of IT including measures such as ease of use, reliability, and response time. *Information quality* refers to the semantic dimension of IT compromising variables like information understandability, output accuracy, and usefulness of information. *IT system performance* is seen as necessary condition for *IT acceptance* in terms of *user satisfaction* and *use*. Whereas *user satisfaction* describes enjoyment and happiness of stakeholders such as managers and employees, *use of a system* is measured in terms of actual usage (use versus non-use), motivation to use the system, and frequency of usage. IT acceptance is seen as necessary condition for *individual impacts*. It refers to how IT contributes to performance of users (e.g. time for task completion) and managers (e.g. time and quality of decisions). *Individual impacts* serve as precondition for desirable *organizational impacts* expressing IT's contribution to innovation, market share, and profits (DeLone and McLean 1992).

The proposed process model was later transferred into a variance model, which was refined and tested empirically by DeLone and McLean (2003). Most essentially, the authors found that beside IT performance in terms of *information quality* and *system quality*, specific characteristics of a system's provider—such as assurance, empathy, and responsiveness—defines *service quality*. *Service quality* served as a third antecedence of acceptance of the IT system (DeLone and McLean 2003).

2.2.2.4 The Melville-Kraemer-Gurbaxani-Model on IT Resources

The latest IT value model was proposed by Melville et al. (2004). Drawing upon an extensive review of previous literature, the authors took resource-based view of the firm (see Sect. 2.2.1) as theoretical lens to describe how IT results in outcomes. In their integrative process model, IT is seen as antecedence for performance. The value generation process starts within an organization referred to as *focal firm*. Within a *focal firm*, IT business value can be realized from a combination of multiple *IT* and related *complementary organizational resources*. *IT resources* are categorized into *technological* and *human resources*. The former one compromises IT infrastructures and business applications within a company. The latter one combines employees' technical (e.g. programming and systems integration) and managerial skills (e.g. collaboration with internal and external units or project

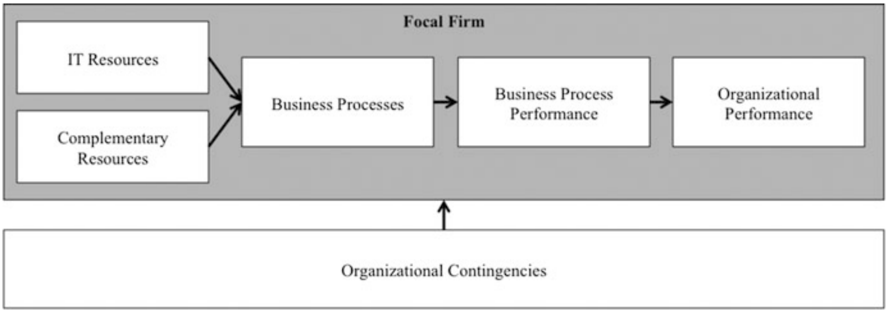

Fig. 2.9 The Melville-Kraemer-Gurbaxani-Model based on Melville et al. (2004)

planning). *Complementary organizational resources* refer to non-IT resources that provide synergies between IT and other firm resources (Melville et al. 2004). As defined by Barney, such resources are classified into physical (e.g. distributed locations and firm's global reach), human (e.g. skills and expertise), and organizational resources (e.g. advanced processes and corporate's structures) (Barney 1991; Nöhren and Heinzl 2012).

According to Melville et al. (2004), the combination of *IT* and *complementary organizational resources* impacts and enables *business processes* as shown in Fig. 2.9. *Business processes* are defined on an abstract level as value generating activities such as sales, manufacturing, and distribution, responsible for transforming sets of inputs into particular outputs. These *business processes* influence performance on two levels. It is differentiated between *business process* and *organizational performance*. The first one refers to measures of operational efficiency of business processes such as quality, inventory turnover, or cycle time. Performance on business process level is described as necessary condition for *organizational performance* in terms of profitability, market value, and competitive advantage (Melville et al. 2004).

In addition, Melville et al. (2004) emphasize on the importance of external *organizational contingencies* shaping the value generation process within a *focal firm*. It is differentiated between factors from the competitive micro environment such as certain industry characteristics and resources of trading partners as well as factors of the macro environment including characteristics on a country level (Melville et al. 2004).

2.2.2.5 Summary and Synthesis of Information Technology Value Models

To sum up, all the four afore-mentioned value models discussed above are concerned with justifying returns of IT in terms of IT-related financial and non-financial investments (Dehning and Richardson 2002; Soh and Markus 1995), IT resources (Melville et al. 2004), or IT systems in general (DeLone and

McLean 1992). None of these models expose the distinct role of a particular IT artefact such as process-centric enterprise systems. However, these four models are particularly valuable in understanding the questions of “where” to find the impact of IT, “how” the relationship between IT artefact (predictor) and its outcome (dependent) variable can be understood, and “what” categories serve as a measure of IT value on an abstract level. Three key contributions can be derived:

First, with respect to the “how”-question and the relationship between predictor and dependent variable, it was found that IT can be seen as an input (Dehning and Richardson 2002; Soh and Markus 1995) or as an antecedence (DeLone and McLean 1992, 2003; Melville et al. 2004) in an IT value generation process. In general, both IT anchorings are useful in IT value research. A criterion-specific perspective is particularly beneficial in measuring transformations of IT-related inputs in desirable outputs (e.g. Bharadwaj 2000; Nöhren and Heinzl 2012). The key conclusion of these studies is an assessment of performance of an internally executed transformation process. However, due to the fact that IT-related inputs are frequently seen as a bundle of variables (e.g. Alpar et al. 2001; Nöhren and Heinzl 2012), their individual impact on a dependent variable cannot be derived by taking a criterion-specific point of view. In contrast to this, a criterion-free anchoring evaluates the impact of a particular IT resource (e.g. Dibbern et al. 2008; Goo et al. 2008) on a particular dependent variable. As outlined above, this research focuses on process-centric enterprise systems and their impact on alignment and performance as dependent constructs. Consequently, enterprise systems are seen as antecedence in the value generation process, which makes a criterion-free anchoring appropriate.

Second, with respect to the level of analysis and the related “where”-question, it was found that IT outcomes could be captured on three levels within a firm. On an individual level, the value of IT artefacts for individual users can be assessed (e.g. DeLone and McLean 1992, 2003; Soh and Markus 1995). This level is particularly helpful when focusing on user-centric IT such as social software and collaboration systems where the outcome is primarily generated on a user level (e.g. Chai et al. 2011; Parameswaran and Whinston 2007; Suh et al. 2011). In contrast to this, the impact of process-centric enterprise software is best understood on a process level. Melville et al. (2004) found that IT’s business contribution occurs in a successive manner. First, in combination with a set of complementary organizational resources, IT artefacts enable business processes (Melville et al. 2004). This is particularly true for process-centric systems such as ERP or CRM. These business processes result in business process performance first, before emerging to organizational level outcomes (Dehning and Richardson 2002; Melville et al. 2004).

Third, on a very abstract level, IT value models expose “what” IT value is. In particular, three key findings can be derived. First, IT generates an *intermediate outcome* that is directly attributed to a certain IT artefact under study. DeLone and McLean (1992, 2003) found that IT systems performance—reflected by the requirements of a firm—results in acceptance in terms of user satisfaction and actual usage in a first stage. In addition, Soh and Markus (1995) found that IT expenditures lead to the creation of IT assets in terms of useful, well-designed, or flexible applications. These *intermediate outcomes* contribute to performance outcomes. However, superior *intermediate outcomes* do not inevitably result in higher performance on business process or organizational level. Several internal and external contingencies shape IT's value generation process (Melville et al. 2004; Soh and Markus 1995). The lower the organizational level of measuring IT's contribution, the closer the link to a particular IT artefact under investigation and the less severe is the role of internal and external contingencies. Second, *intermediate outcomes* are frequently observed by the means of "fit". For instance, Soh and Markus speak of useful and well designed application in their notion of IT assets and highlight that IT must be "*(...) designed in such a way that it fits the firm's task effectively (...)*" (Soh and Markus 1995: 30). In addition, DeLone and McLean (1992, 2003) measure the *intermediate outcome* of software systems in terms of systems quality and information quality and how these components fit with user and organizational requirements. Third, when looking at IT's contribution to process and organizational performance, all four models discussed above point out that appropriate measures needs to be designed in close relationship to those potential values that can be realized by this particular IT artefact. Melville et al. defined business process performance as "*(...) a range of measures associated with operational efficiency enhancement within specific business processes, such as quality improvement of design processes and enhanced cycle time within inventory management processes*" (Melville et al. 2004: 296). According to Dehning and Richardson, "*business process performance measures include gross margin, inventory turnover, customer service, quality, efficiency, and other cost, profit margin, and turnover ratios*." (Dehning and Richardson 2002: 9). Organizational performance can broadly spoken be operationalized in terms of a set of market-related measures such as return of investments, profitability, or market share (Dehning and Richardson 2002; DeLone and McLean 1992; Melville et al. 2004; Soh and Markus 1995). Drawing upon these findings, a more generic model of IT value as presented in Fig. 2.10 can be derived.

It can be deduced that each IT artefact such as enterprise software generates a certain *intermediate outcome*. This outcome is directly related to the artefact under investigation and can be measured in terms of its fit with the requirements of a firm. *Intermediate outcomes* impact *business performance outcomes* in terms of business process and organizational performance in two consecutive steps. The link between outcome and IT system becomes fuzzier from *intermediate outcomes* through *business process performance* to *organizational performance* due to an increasing influence of external and internal macro and micro environmental contingencies shaping value generation process. Previous studies frequently showed that firms might see internal process improvements from IT, but that the same gains do not emerge to a firm level (Mittal and Nault 2009). Consequently, this study focuses on *intermediate outcomes* in terms of software fit and its impact on *business process performance*. Organizational level outcomes were not investigated. The next subsection gives a definition of the performance outcome concepts applied in this study.

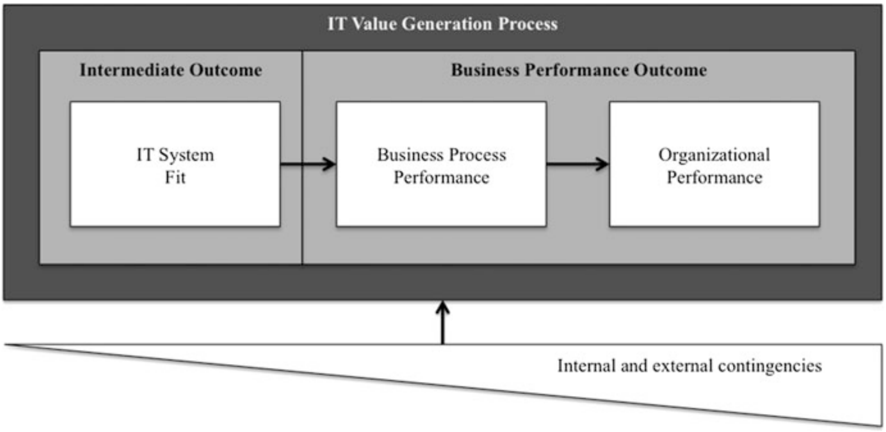

Fig. 2.10 A generic model on IT value generation process

2.2.3 Previous Contribution on Software and Sourcing Performance

Based on the generic model of IT value generation, this section continues with a discussion of literature related to software and sourcing performance. Previous findings are classified with respect to their research stream into studies on process-centric software performance (Sect. 2.2.3.1), software sourcing performance (Sect. 2.2.3.2), and IT outsourcing performance (Sect. 2.2.3.3). Findings are summarized in Sect. 2.2.3.4 and dependent concepts of this study are introduced. Table 2.2 gives an overview of previous contribution on software and sourcing performance.

2.2.3.1 Studies on Process-Centric Software Performance

Studies on process-centric software performance typically investigated performance of specific types of enterprise systems. Two studies that focused on performance outcomes of packaged ERP systems on an **organizational level** were identified. Cotteleer and Bendoly (2006) conducted a multimethod approach including interviews and reviews of several internal documents in order to assess the contribution of ERP implementation on operational performance within a case company. It was found that the investigated enterprise system contributes to *near-term operational performance improvements* (such as lead-time) and *long-term improvements* (such as learning effects) (Cotteleer and Bendoly 2006). In another study on ERP implementation, Ranganathan and Brown (2006) analysed enterprise system's impact on *abnormal returns*. In particular the effects of application's *functional* and *physical scope* as well as *vendor's reputation* were investigated. *Functional scope* refers to the types of ERP modules implemented. In particular it

Table 2.2 Previous contribution on software and sourcing performance

Research stream	Focus of investigation	Performance concepts	Impact level	Sources
Process-centric software performance	Packaged software	Operational performance	Organization	Cotteleer and Bendoly (2006)
		Market performance	Organization	Ranganathan and Brown (2006)
		Operational, organizational, strategic, technological performance	Business process IT function Organization	Grant (2003)
		Infrastructure, managerial, operational, organizational, strategic performance	Business process IT function Organization	Shang and Seddon (2002)
		Operational efficiency	Business process	Banker et al. (1990)
	In-house software	Financial performance, usage	Business process	Houdeshel and Watson (1987)
Software sourcing performance	On-demand software performance	IT-enabled innovation	Organization	Malladi and Krishnan (2012a, b)
		Net benefits	Organization	Walther et al. (2012)
		System quality and system effectiveness	Business process	Hsieh and Huang (2012)
		Orientation towards innovation	IT function	Malladi and Krishnan (2012a, b)
		Perceived performance, satisfaction	IT function	Susarla et al. (2003)
		Performance outcomes	IT function	Susarla et al. (2009)
	On-demand, on-premises software performance	Software quality	IT function	Choudhary (2007a, b)
		Profit (vendor)	IT function	Zhang and Seidmann (2010)
	In-house software performance	Effectiveness of functionalities	Organization	Banker and Kemerer (1992)
		Operational efficiency, responsiveness, flexibility	Business process	Nidumolu (1995)
		Outsourcing success (overlap with SET satisfaction)	IT function	Wang (2002)
IT outsourcing performance	Sourcing success	Overall satisfaction	Business process IT function Organization	E.g. Koh et al. (2004), Nöhren and Heinzl (2012), Saunders et al. (1997)
		SET satisfaction	Business process IT function Organization	E.g. Goo et al. (2008), Grover et al. (1996), Lee and Kim (1999), Lee et al. (2004), Saunders et al. (1997)

(continued)

Table 2.2 (continued)

Research stream	Focus of investigation	Performance concepts	Impact level	Sources
		Extended SET satisfaction	Business process IT function Organization	Lacity and Willcocks (2001)
	Project performance	Cost, duration, quality	IT function	Gopal et al. (2002)
		Contract performance	IT function	Domberger et al. (2000)
		(Extra) Costs	IT function	Dibbern et al. (2008)
		Project performance	IT function	Nidumolu (1995)
	Process performance	Perceived business process improvements	Business process	Downing et al. (2003)
	Firm performance	Abnormal returns	Organization	Agarwal et al. (2006), Gewald and Gellrich (2007)
		Long-term business impact	Organization	Mojsilović et al. (2007)
		Financial metrics	Organization	Wang et al. (2008a)

was differentiated between enterprise support modules (e.g. human resources) and value-chain modules (e.g. sales and distribution). *Physical scope* refers to number of sites (regional or international) an ERP system was implemented. It was found that both, functional and physical scope positively impact abnormal returns (Ranganathan and Brown 2006).

Two studies combined **IT function, process and organizational level** measures with each other. Both studies provide categorizations of items measuring firm performance, business process performance, and IT function performance. In an investigation of strategic alignment Grant (2003) conducted an in-depth exploratory single case study to examine the success of the introduction of an ERP system within a case company. In particular, the implementation of a global financial accounting and reporting system was investigated. This study measured ERP success in terms of its *strategic*, *organizational*, *operational*, and *technological impact*. *Strategic impact* (organizational level) refers to the support of long-term initiatives such as *strategic alliances* and *market opportunities*. *Organizational impact* (organizational level) describes enterprise-wide benefits like *coordination between and across business units*. *Operational impacts* (business process level) are efficiency gains in terms of *improved quality* and *improved decision support capabilities*. At last, *technological impact* (IT function level) was defined by the inherent characteristics of an application such as *standardized IT operations* and *improved processing power* (Grant 2003). Drawing upon a review of vendorreported success stories combined with interviews with IT managers, Shang and Seddon (2002) developed a list of 21 indicators to assess performance of packaged ERP software on process and organizational level. These indicators are grouped into five dimensions, *operational*, *managerial*, *strategic*, *IT infrastructure*, and *organizational benefits*. *Operational benefits* (business process level) such as *cost reduction and productivity improvements* refer to software's impact on day-to-day activities. On a *managerial dimension* (business process level), measures such as *better resource management* and *improved decision making and planning* assess application's contribution to support activities of line managers. On a *strategic dimension* (organizational level), Shang and Seddon identified indicators for an enterprise system's impact on competitive advantages like *support for business growth* and *building external linkages*. *IT infrastructure benefits* (IT function level) refer to reusable and shareable IT infrastructures for multiple business applications. Finally, *organizational benefits* (organizational level) describe companywide improvements such as *empowerment* and *changing work pattern* (Shang and Seddon 2002).

Two early studies on software performance evaluated the impact on a **process level**. Houdeshel and Watson (1987) conducted a single case study to assess benefits and success of an internally developed management information and decision support system (MIDS). In their study, the in-house application led to multiple paybacks such as an improved communication among internal and external stakeholders and improved information quality. The success of the system was measured in terms of a *cost-benefit analysis*, *frequency of use*, and *user satisfaction* (Houdeshel and Watson 1987). Banker et al. (1990) conducted a study on *operational efficiency* of a point-of-sale and order-coordination technology (POS) within an American fast food company. By applying data envelopment analysis and non-parametric hypotheses testing, the authors benchmarked the efficiency of restaurants that had implemented the POS system with those that had not. It was found that the packaged software contributed to the efficiency of the sites. In their study, *operational efficiency* was defined as the effects of IT investments on a business process level (Banker et al. 1990).

To sum up, in the light of a set of internal and external contingencies shaping software's value contribution performance outcome (e.g. Dehning and Richardson 2002; Melville et al. 2004; Soh and Markus 1995), a thorough performance outcome measure needs to focus on a process level (Melville et al. 2004). Against this background, studies on organizational level impact of process-centric software (Cotteleer and Bendoly 2006; Ranganathan and Brown 2006) are not further investigated. Banker et al. (1990) as well as Houdeshel and Watson (1987) relied on a broad definition of business process performance in terms of operational efficiency, financial performance, and usage. These concepts were not specified in greater detail by the authors. Consequently, these studies are dropped from further consideration. The most comprehensive conceptualization of process-centric software performance was found by Grant (2003) as well as Shang and Seddon (2002). The authors defined a set of indicators for measuring perceived business process performance from an operational and managerial perspective.

2.2.3.2 Studies on Software Sourcing Performance

Most studies related to software sourcing performance focus on **on-demand software**. Among them, two papers were identified considering organizational level impacts. In a study on IT-enabled innovation, Malladi and Krishnan (2012a, b) analysed the impact of on-demand software adoption on *innovation* moderated by firm's past *outsourcing experience*, *internal IT architecture flexibility*, and *process management maturity*. The dependent variable was coded as binary variable specifying whether an organization has patented any IT architectures, products, services, or processes within the past year. A positive relationship between on-demand software usage and innovation was found (Malladi and Krishnan 2012a, b). In order to develop an updated DeLone-McLean IS success model, Walther et al. (2012) conducted literature review to identify success factors and value propositions in software-as-a-service research. Based upon an investigation of 36 papers, the authors classified most salient on-demand outcomes based on their number of usage in previous research. Ten variables such as *cost savings*, *cost flexibility*, and *mobility* had been identified as appropriate for measuring *net benefits* of software-as-a-service (Walther et al. 2012).

Three papers were identified studying impacts of on-demand software related to a corporate's IT function. Malladi and Krishnan (2012a, b) investigated the impact of cloud computing on *CIO's strategic focus*, which was defined as direction towards innovation and new product development capabilities. A positive relationship moderated by *process management maturity* and *internal coordination IT capabilities* was found (Malladi and Krishnan 2012a, b). In a study on application service provisioning, Susarla et al. (2003) tested factors impacting client's satisfaction with its service vendor. Drawing upon expectation disconfirmation theory, the influence of *perceived performance* was found to be highly significant in the derived path model. *Perceived performance* was defined as five-item construct including measures related to vendor performance (*better maintenance support*, *ability to implement IT solutions rapidly*, and *access to best technology*) and application performance (*ability to integrate information from various functional applications* and *low implementation and service costs*) (Susarla et al. 2003). In a later study on application service provisioning, Susarla et al. (2009) investigated the impact of contracting mechanisms on performance outcomes based on transaction cost economics. In particular, it was found that those contracts, that are not aligned with transaction attributes resulted in budget overruns and achieved lower performance (Susarla et al. 2009).

In conceptualizing *user satisfaction* with on-demand CRM, Hsieh and Huang (2012) differentiated between *system quality* and *process performance*. The former one refers to *perceived usefulness* and *ease of use* of an application. *Process performance* was proposed to be a measure of *system effectiveness* in terms of an assessment of the communication, marketing, and relationship performance of the CRM process. The authors emphasized the mediating role of *intention to use* in realizing *process performance outcomes* (Hsieh and Huang 2012).

Three studies were found that simultaneously address **on-premises and on-demand software** at the level of the IT function. Those studies were conceptual-mathematical in nature. In two studies by Choudhary (2007a, b) a model to compare software quality in a software-as-a-service setting with those in an on-premises mode was developed and refined. Taking a vendor's perspective, its intention to invest in product quality was mathematically deduced. In the proposed two-period model, *software quality* is seen as a function of time. Investments in software development and improvement were found to be more likely in on-demand settings. This should result in a higher quality of software-as-a-service compared to on-premises applications (Choudhary 2007a, b). A third mathematical model was introduced by Zhang and Seidmann (2010). Drawing upon specific characteristics of on-demand and on-premises licencing in terms of *uncertainty about quality and compatibility of future updates*, *network externalities*, and *provider's ability to commit to future prices*, the optimal mode to offer software from a monopolistic vendor's perspective was discussed. In a two-period conceptual model, optimal licensing strategy is mainly impacted by quality uncertainty and network effects (Zhang and Seidmann 2010).

In-house software was observed on three levels. On an organizational level, Banker and Kemerer (1992) introduced a non-empirical mathematical principal-agent model for performance evaluation of customized external software development projects. Taking a client's perspective, four measures for project outcomes, *short-term costs* (initial development cost), *long-term costs* (maintenance costs), *short-term benefits* (timeliness of software development), and *long-term benefits* (effectiveness) were derived (Banker and Kemerer 1992). On a business process level, Nidumolu (1995) studied the effects of vertical (coordination initiated by project managers) and horizontal coordination mechanisms (mutual communication between users and IT staff) on *project performance* in customized software development settings. In this study, *project performance* was measured in terms of *process* and *product performance*. The former one describes how well the software development process was conducted in terms of *learning*, *control*, and *quality of interactions*. The latter one refers to the performance of the delivered system by using measures for *operational efficiency* (e.g. *reliability of software*, *cost of operations*), *responsiveness* (e.g. *ease of software use*, *ability to customize outputs to various user needs*), and *flexibility* (e.g. *cost of adapting software to changes in business*, *cost of maintaining software over lifetime*) (Nidumolu 1995). Finally, performance of a corporate's IT function was studied by Wang (2002). Drawing upon transaction cost economics, a multiple regression analysis was conducted to assess the effect of the exogenous variables *contractor reputation*, *asset specificity*, and *uncertainty* on the mediator *post-contractual opportunism* and the dependent variable *outsourcing success*. In this study, success was operationalized as six-item reflective construct. All items, *focus on core business*, *increase IS competence*, *access to skilled personnel*, *cost savings on human resources*, *cost savings on technological resources*, and *control of IS expenses* are closely related to SET satisfaction which will be discussed in Sect. 2.2.3.3 (Wang 2002).

To sum up, several studies related to this stream of research are either mathematical (e.g. Banker and Kemerer 1992; Zhang and Seidmann 2010) or conceptual (e.g. Hsieh and Huang 2012; Walther et al. 2012) in nature. For instance, taking a vendors perspective, on-premises and on-demand licensing is compared and implications for software quality is derived (Choudhary 2007a, b). Due to the fact that an empirical study is conducted in this work, conceptual and mathematical papers are not further taken into consideration. In addition, organizational level studies (e.g. Malladi and Krishnan 2012a, b) were dropped. The above outlined business process performance conceptualization of Grant (2003) as well as Shang and Seddon (2002) encompasses the definition of Nidumolu (1995). Consequently, this study adds little value to the business process performance concept and is not further discussed. The remaining studies related to this stream of research stress the importance of considering sourcing performance of software related to a corporate's IT function (e.g. Susarla et al. 2003, 2009; Wang 2002). In order to enrich the discussion of this concept, previous studies on outsourcing performance are discussed in the following subsection.

2.2.3.3 Studies on IT Outsourcing Performance

As outlined above, the notion of performance is typically vague in nature (Dibbern et al. 2004). Looking at more than 20 years of IT outsourcing research, a huge body of literature deals with the outcomes of sourcing arrangements for client firms. Over the years, scientists adopted a wide dispersion of dependent concepts (Dibbern et al. 2004; Mahnke et al. 2005). In this section, previous findings are classified based on their focus of investigation into studies on sourcing success, project performance, process performance, and organizational performance. It has to be noted that a complete discussion of dependent constructs applied in IT outsourcing research is beyond the scope of this work. Interested readers may refer to works of Dibbern et al. (2004) and Lacity et al. (2010).

Sourcing success is most frequently operationalized in terms of satisfaction (Dibbern et al. 2004) with perceived benefits derived from IT outsourcing arrangements (e.g. Grover et al. 1996; Koh et al. 2004; Lacity and Willcocks 1998; Lee et al. 2004; Wang 2002). Lacity et al. (2010) found that sourcing success is the predominant performance construct in IT outsourcing research. In sum, more than 170 out of 376 investigated studies relied on outsourcing success measures (Lacity et al. 2010). Satisfaction is defined as "*a positive affective state resulting from the appraisal of all aspects of a firm's working relationship with another firm*" (Anderson and Narus 1984: 66). Previous studies relied on overall measures of (perceived) outsourcing success in terms of satisfaction with a sourcing arrangement in general and the intention to retain a particular service vendor (e.g. Koh et al. 2004; Saunders et al. 1997). Other studies added more granularity to the discussion. For instance, Lacity and Willcocks (2001) categorized such perceived benefits in terms of *financial restructuring*, *core competences*, *technological catalyst* (e.g. greater flexibility in technology), *business transition* (e.g. support of organizational change), *business innovation* (e.g. innovation of processes and skills), and *new markets* (e.g. joint ventures). This notion was eventually limited to fewer factors in quantitative performance measurement. Lee et al. (2004) identified SET satisfaction in terms of *cost efficiency* (former *financial restructuring*), *technological catalysis*, and *strategic competence* (former *core competence*) as most salient for IT outsourcing success research. *Strategic competence* refers to "*redirecting the business and IT into core competencies*" (Lacity and Willcocks 2001: 316). It is a measure of outsourcing's contribution to refocus on core competences, increasing IT competence, access to skilled IT personnel, as well as a changed focus on strategic activities (Grover et al. 1996; Lee et al. 2004; Saunders et al. 1997). *Technology catalysis* refers to "*strengthening resources and flexibility in technology service to underpin business' strategic direction*" (Lacity and Willcocks 2001: 317). It evaluates outsourcing's contribution to reducing risks of technological obsolescence, to transform traditional IT infrastructures, to access key technologies, to incorporate new technologies and skills, as well as to increase technological flexibility (Grover et al. 1996; Lacity and Willcocks 2001; Lee et al. 2004; Saunders et al. 1997). Finally, *cost efficiency* refers to "*improving the business' financial position*" (Lacity and Willcocks 2001: 315). It captures the avoidance of major capital expenditures, the increases in economies of scale in human and technological resources as well as the extent of greater control over IT-related expenditures (Grover et al. 1996; Lee et al. 2004; Saunders et al. 1997).

Project performance refers to outcomes directly attributed to a particular sourcing arrangement. In contrast to success measures, which typically rely on Likert-scale scores (Dibbern et al. 2004), project performance measures are frequently quantifiable in nature. For instance, Gopal et al. (2002) evaluated performance of outsourcing software development projects in terms of *time elapsed* (project duration), *quality* (software rework), and *costs* (effort). It was found that these three concepts are interdependent with each other. Domberger et al. (2000) evaluated the realization of expectations from outsourcing contracts. This contract performance was captured in terms of the eight items *service availability and timeliness*, *out-of-hours availability*, *response in emergencies*, *provision at expected cost*, *delivery to expected quality*, *accuracy of advice*, *correctness of error fixes*, and *minimization of system downtime*. Dibbern et al. (2008) measured performance of offshored software development and maintenance projects by the means of extra costs. These expenditures in terms of *control*, *coordination*, *design*, and *knowledge transfer* are costs that occur on top of a particular outsourcing arrangement. In addition, Nidumolu (1995) evaluated performance of software development projects by measuring *project performance* (labelled as *process performance*) in terms of learning, control, and quality.

Process performance is defined as "the level of performance of a process, such as process costs, operational efficiency, quality, or level of customer satisfaction" (Lacity et al. 2010: 426). In a study by Downing et al. (2003) the perceived business process improvements of IT outsourcing by business process managers was investigated.

Firm performance refers to "the degree to which a client organization reports business performance improvements as a result of an outsourcing decision, such as stock price performance, return on assets, expenses, and profits" (Lacity et al. 2010: 425). For instance, Agarwal et al. (2006) conducted an event study to analyse the impact of e-business outsourcing announcements on abnormal returns. In another event study, Gewald and Gellrich (2007) measured the impact of outsourcing statements on organizational performance. In their event study, performance was measured in terms of market reaction. The influence of outsourcing-specific risks such as transaction size, contract length, and outsourcing experience was investigated. Mojsilović et al. (2007) developed a model for analysing IT outsourcings' long-term impact on client's business. In addition, Wang et al. (2008a) emphasized the importance to include process level performance metrics such as sales per employee, total sales, and expenses in performance measurement. Performance on this level may be transformed into firm-level outcomes. In measuring performance on firm level, the authors relied on measures such as return on assets and return on investment (Wang et al. 2008a).

To sum up, a huge body of research on IT outsourcing outcomes exists. Concepts related to firm performance are concerned with capturing outcomes on an organizational level in terms of abnormal returns or other financial measures (e.g. Agarwal et al. 2006; Gewald and Gellrich 2007; Wang et al. 2008a). Variables related to this concept are particularly useful when investigating the position of certain companies within a competitive environment. As have been outlined, this study focused on process level performance outcomes. Consequently, firm performance is beyond the scope of this investigation. Concepts related to project performance determined the value of specific IT outsourcing tasks (e.g. Dibbern et al. 2008; Domberger et al. 2000; Gopal et al. 2002). Such tasks have an inherent duration with an explicit or an implicit completion date. Studies in this area rather focused on the realization of tasks related to the IT function and their impact on process or organizational value than on the performance of a particular enterprise system itself. Consequently, project performance concepts are less relevant for this research work.

The identified study on process performance by Downing et al. (2003) shows great overlap with the perceived performance concepts of Grant (2003) as well as Shang and Seddon (2002). However, the latter ones were found to be more explicit in defining a range of indicators identifying the contribution of process-centric software on business processes. Therefore, the concept by Downing et al. (2003) is not further investigated.

In contrast to this, the remaining sourcing success measures are more essential for the endeavour of this study. Most frequently IT outsourcing performance studies relied on the notion of *sourcing success* in terms of SET satisfaction (Goo et al. 2008; e.g. Grover et al. 1996; Lee et al. 2004). SET satisfaction provides a comprehensive measure to evaluate perceived performance from a strategic, economic, and technological point of view. It is frequently used to evaluate sourcing outcomes related to a corporate's IT function (e.g. Lacity and Willcocks 2001; Lee et al. 2004; Saunders et al. 1997). Against this background, SET satisfaction is identified as most valuable outcome concept for measuring sourcing performance of software.

2.2.3.4 Summary and Definition of Performance Outcome Concepts

This study focuses on a specific type of software artefacts. In particular, process-centric enterprise systems like ERP and CRM systems are investigated (Grant 2003; Shang and Seddon 2002; Strong and Volkoff 2010). Such software artefacts generate their primary impact on corporate business processes but also generate a strong secondary effect on a corporate's IT function (Swanson 1994) in terms of system development, system operation, and management (Heinzl 1993). Consequently, performance measurement has to capture both impacts of process-centric enterprise systems.

Drawing upon the literature review, two performance concepts were identified. First, business process performance assesses software's value within the business units (Grant 2003; Shang and Seddon 2002). The definition of this concept is given in Table 2.3. Second, sourcing performance in terms of SET satisfaction determines enterprise system's impact on the IT function (e.g. Goo et al. 2008; Grover et al. 1996; Lacity and Willcocks 2001; Lee and Kim 1999; Lee et al. 2004; Saunders et al. 1997). It measures the strategic, economic, and technological benefits of software. The definition of the three subconcepts related to software sourcing performance is given in Table 2.4. By including two concepts within this study, both, the primary value within the business units and the secondary impact on corporate's IT function can be investigated.

2.2.4 Previous Contribution on Software Alignment

In the presented study, software alignment is measured in terms of fit between software and business process (Strong and Volkoff 2010). Misfits occur as "*(. . .) the result of differences between the structures embedded in the package and those embedded in the organization*" (Soh and Sia 2004: 375). From a RIT perspective, such misfits are cases where elements of the real world are not adequately represented by an application (Sia and Soh 2007). In previous research, three perspectives of fit between software and business process structures emerged

Table 2.3 Definition of business process performance

Construct	Definition	Sources
Business process performance	Contribution of process-centric software to business process performance in terms of operational and managerial benefits identified by Grant (2003) as well as Shang and Seddon (2002)	Grant (2003), Shang and Seddon (2002)

Table 2.4 Definition of sourcing performance

Construct	Definition	Sources
Strategic benefits	A client organization's degree of satisfaction with software in terms of redirecting IT function into core competencies	E.g. Goo et al. (2008), Grover et al. (1996), Lacity and Willcocks (2001), Lee and Kim (1999), Lee et al. (2004), Saunders et al. (1997)
Economical benefits	A client organization's degree of satisfaction with software in terms of improving cost position of the IT function	
Technological benefits	A client organization's degree of satisfaction with software in terms of strengthening IT function's technological flexibility	

(Maurer et al. 2012; Nöhren et al. 2014). These three notions are discussed subsequently in order to derive an appropriate concept to evaluate software-business process fit.

2.2.4.1 The Sia-Soh-Notion of Software-Organization Fit

The first one is a taxonomy developed by Sia and Soh (2007). Based on institutional theory and RIT, four types of software misfits were derived (see Fig. 2.11). According to Sia and Soh (2007), misfits arise from differences in the actual organizational structures and the structures anticipated by a software developer. Drawing upon the institutional theory organizational structure can either be imposed or voluntarily acquired (Scott 1987). Imposed structures result from external sources such as the authoritative of a government (laws and regulations) and specific industry characteristics necessary to remain and perform in business (DiMaggio and Powell 1983; Soh and Sia 2004). Voluntary structures are deliberately adopted by an organization over time (Soh and Sia 2004). These structures arise from internal contingencies and encompass organizational norms and routines associated with efficient and effective resource acquisition and utilization (Scott 1987). Voluntarily acquired structures distinguish firms from each other (Soh and Sia 2004).

By including a RIT perspective, Sia and Soh (2007) enriched their discussion of imposed and voluntarily acquired structures by further differentiating between deep and surface structures of the software artefact. Thereby, four types of software-organization misfits were derived. First, *imposed deep structure misfits* refer to deficiencies of an enterprise system in capturing central elements of an organizational reality resulting in reduced levels of operational efficiency. Second, *imposed surface structure misfits* are less severe but require workarounds when operating the

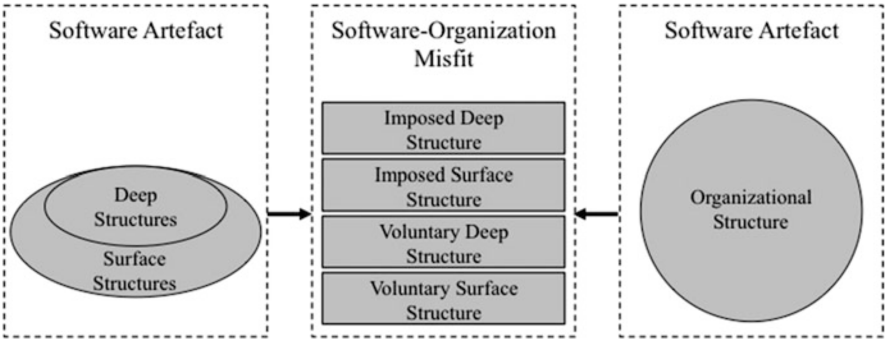

Fig. 2.11 Software-organization-misfit typology by Sia and Soh (2007)

user interface. Third, *voluntary deep structure misfits* result from the idiosyncratic processes of an organization conflicting with the core of an enterprise system. Finally, *voluntary surface structure* misfits are related to inappropriate or missing access, input, or output of software packages conflicting with organizational idiosyncrasies (Sia and Soh 2007).

2.2.4.2 The Strong-Volkoff-Notion of Software-Organization Fit

A second taxonomy was proposed by Strong and Volkoff (2010). Based on grounded theory, the authors identified six sources of software misfits—functionality, data, usability, role, control, and organizational culture—embedded within an enterprise system (see Fig. 2.12). In their study, surface structure was defined by software usability. *Usability misfits* occur, when the interaction with an enterprise system is confusing or inconvenient. Deep structures were defined by software functionality and data representation. *Functionality misfits* arise when particular functionalities are needed but missing. *Data misfits* occur, when data stored in an enterprise system leads to reduced quality in terms of timeliness, inaccessibility, or inappropriateness. Beside a deep and surface structure dimension, it was found that misfits also arise from latent structures. These latent structures, in terms of organizational culture, role, and control, are not directly designed within an application—like deep and surface structures—but arise from them as second order structures. Roles define the responsibilities of people within an organization. If the roles embedded in an enterprise system are inconsistent with the organizational responsibilities, *role misfits* occur. *Organizational culture misfits* result from an application that requires ways of working contrary to organizational norms. Finally, *control misfits* arise, when the control carried out by the system provides too much or too little opportunities (Strong and Volkoff 2010).

Strong and Volkoff (2010) further differentiated these sources of misfits into *deficiencies* and *impositions*. The former one refers to problems that arise when software features are needed but missing. The latter one specifies those

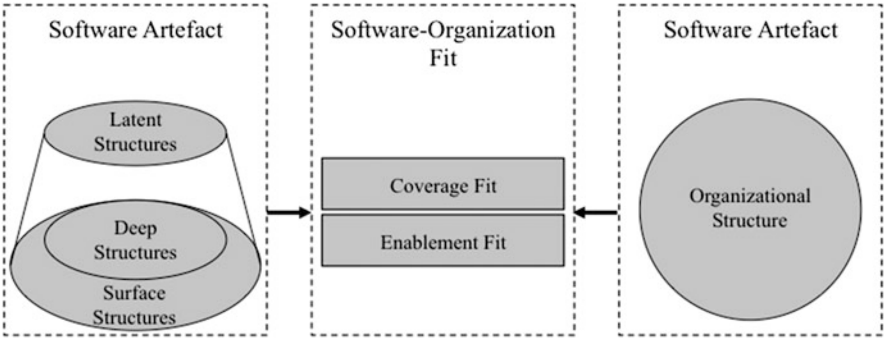

Fig. 2.12 Software-organization-fit typology by Strong and Volkoff (2010)

complications that result from the inherent structure of software that requires ways of working conflicting with the organizational reality. Drawing upon this classification, two fit constructs were developed. The first one, referred to as **coverage fit**, means that an application is free from functionality, data, usability, role, and control deficiencies. The second construct, termed **enablement fit** emerges from the absence of functionality, data, usability, role, control, and organizational culture impositions. Organizational culture misfits can only occur as impositions. Furthermore, enablement fit describes a situation where a company is better off with the existing system than with its legacy system (Strong and Volkoff 2010).

2.2.4.3 The Nöhren-Heinzl-Kude-Notion of Dynamic Software-Business Process Fit

Drawing upon a RIT perspective, IT is seen as representation of a corporate reality. Consequently, in the light of process and software structure change (e.g. Maltz and Kohli 1996; Nissen and Burton 2011; Nöhren et al. 2014; Zajac et al. 2000), software alignment must be seen as non-deterministic and dynamic process (Sabherwal et al. 2001; Vessey and Ward 2013). In particular, both, software as well as corporate's business processes must be adaptive in nature in order to coevolve with each other (Benbya and McKelvey 2006; Vessey and Ward 2013).

In general, change can be seen as a function of magnitude, direction, and timing of shifts in business process and software structures (Nissen and Burton 2011; Nöhren et al. 2014; Zajac et al. 2000). *Magnitude* expresses the extent of transformation. It describes whether business process or software reality experiences an incremental or a radical shift (Luo and Strong 2004; Orlikowski 1993). *Direction* expresses the course of an transformation. It describes to the path and the content of business process or software structure change. Finally, *timing* refers to date and frequency of transformation. It explains whether periods of stability with no change are rather long or short (Sabherwal et al. 2001).

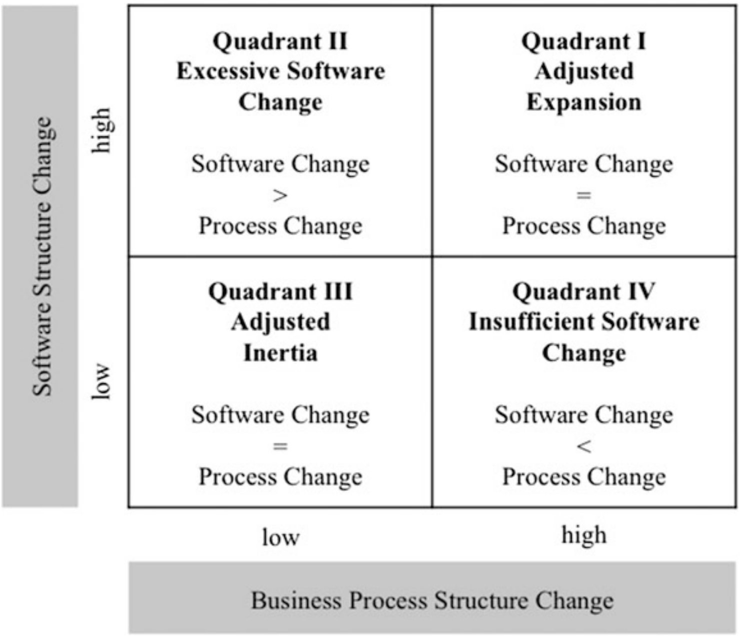

Fig. 2.13 Nöhren-Heinzl-Kude notion of dynamic software-process fit

If software structure change occurs in line with business process structure change, dynamic fit is established. In contrast to this, if software structure change is greater than business process structure change, or vice versa, companies suffer from dynamic misfit. These situations are illustrated in Fig. 2.13 (Nöhren et al. 2014; Zajac et al. 2000).

Figure 2.13 defines two ways to establish dynamic fit. Quadrant I describes the situation in which both, business process structures and software structures experience a high level of change resulting in a beneficial stage for a company. This situation is further referred to as *adjusted expansion*. Quadrant III represents the contrary situation wherein both, business process and software faces little or no change in their structures. A company benefits from persistence. This stage is branded *adjusted inertia*.

Additionally, there are two ways to suffer from dynamic misfit. Quadrant II represents the situation wherein software structure change is high but business processes fail to respond adequately. For organizations in this category, the nonoccurrence of such necessary business process change may be a result of either an organizational inability or unwillingness to change, or business process structures do not suggest the need to change (Zajac et al. 2000). This situation is further referred to as *excessive software change*. Quadrant IV describes the contrary situation in which business process structure change is high but software fails to respond adequately. It captures a condition wherein software has become obsolete, outdated, or otherwise inappropriate in light of business process transformation.

Dynamic misfit is the consequence of the nonoccurrence of required software structure change. This situation is labelled *insufficient software change*.

2.2.4.4 Summary and Development of Software-Business Process Fit Measure

Having outlined three recently published notions of software-business process fit, this section continues with a development of a fit measure for the endeavour of this study. In comparing the fit concepts of Strong and Volkoff (2010) with those of Sia and Soh (2007), the following conclusions can be made. First, the two studies differ with respect to their classification of misfit. Sia and Soh (2007) grouped software misfits with respect to their environmental source. It was found that while voluntarily acquired organizational structure misfits are less severe, companies suffer if imposed structures are not adequately mirrored by their software artefact. Sia and Soh (2007) did not further elaborate what deep and surface structures misfits are. In contrast to this, Strong and Volkoff (2010) added more granularity to the discussion of the software artefact. In particular, the "black box" of deep and surface structures is opened and the authors identified six types of misfits. Second, Strong and Volkoff (2010) provided a novel understanding of the enterprise software artefact. This definition accounts for an incorporation of latent structures that are not directly designed within an application but emerge as second-order structures from deep and surface elements. Third, the typology of Sia and Soh (2007) primarily focuses on why software-business process misfits emerge from environmental forces and how to overcome imposed and voluntarily acquired deep and surface structure misfits.

Against this background, the notion of Strong and Volkoff (2010) appears to be more appropriate to capture a holistic picture on software-business process fit. However, the derived fit constructs show two significant weaknesses. First, it has to be noted that coverage and enablement fit are not free of overlap. In particular, deficiencies—related to the coverage fit construct—"*are problems arising from [software] features that are missing but needed. Empirically, these problems take the form of actions users cannot take because the [enterprise system] is missing functionality, data fields, controls, etc., necessary for those actions.*" (Strong and Volkoff 2010: 737). Impositions—related to the enablement fit construct—"*are problems arising from the inherent characteristics of [software] such as integration and standardization. (...) Empirically, impositions take the form of the [software structure] requiring ways of working that are contrary to organizational norms and practices.*" (Strong and Volkoff 2010: 737). Consequently, deficiencies and impositions often emerge from the same sources in terms of functionality, data, usability, role, and control misfits. Second, enablement fit "*means the [software] permits and enables the organization to operate more effectively, and users to do their work more efficiently, than was the case without [the software]. (...) the organization is better off as a result of implementing the [software]*" (Strong and Volkoff 2010: 746). While coverage fit is independent from a performance anchor, enablement fit

Table 2.5 Definition of dynamic fit

Construct	Definition	Sources
Dynamic fit	Degree to which software is free from deficiencies in terms of functionality, data, usability, role, and control misfits	Based on Nöhren et al. (2014), Strong and Volkoff (2010)

takes a criterion-specific perspective (see Sect. 2.2.2 for a definition of performance anchoring). Based on this discussion, only coverage fit is taken into further consideration for measuring the fit between software and business process structure.

Neither the concepts of Strong and Volkoff (2010) nor those of Sia and Soh (2007) take changing business process or software structures into account. This shortcoming was addressed by the notion of dynamic fit of Nöhren et al. (2014) (see Fig. 2.13). If a company suffers from dynamic misfit in terms of excessive or insufficient software change, software and business process structures are drifting apart. In this situation, an organization experiences deficiencies by the means of functionality, data, usability, role or control misfits. In contrast to this, if company realizes dynamic fit in terms of adjusted expansion or inertia, business process and software structures coevolve with each other. In such conditions, organizations benefit from coverage fit. The definition of dynamic fit is given in Table 2.5. In order to understand how dynamic fit is generated Sect. 2.2.5 investigates concepts related to the dynamic fit process.

2.2.5 The Dynamic Alignment Process

Dynamic fit was defined taking a RIT perspective (Nöhren et al. 2014). Both, RIT and institutional theory sees an organization as a set of structures (Scott 1987; Wand and Weber 1990). These theoretical lenses are closely linked to each other but differ with respect to their focus. While RIT emphasizes on the structures of particular IT artefacts in relationship to other structures of a firm (Sia and Soh 2007; Strong and Volkoff 2010; Wand and Weber 1990), institutional theory takes a more holistic, aggregated, and dynamic view on organizational structures in general (including the IT artefact) and their relationship to changing environmental conditions (Berente and Yoo 2012; Jones and Karsten 2008; Scott 1987). Consequently, by linking RIT with institutional theory, shifts in software and business process structures that result in dynamic fit can be conceptualized.

This section investigates the process by which dynamic fit is established. As outlined above, software alignment must be seen as a non-deterministic and dynamic process (Sabherwal et al. 2001; Vessey and Ward 2013). "*Alignment is not desirable as an end in itself since the business must always change*" (Chan and Reich 2007: 298).

Dynamic fit refers to the coevolution between software and business process structures over time (Nöhren et al. 2014). A change in corporate's business process structures must go along with software structure change and vice versa in order to benefit from adjusted expansion or adjusted inertia (Nöhren et al. 2014; Sabherwal et al. 2001; Zajac et al. 2000). Consequently, dynamic fit possesses two necessary conditions: software structure change and business process structure change (Nöhren et al. 2014).

2.2.5.1 Software Structure Change

Starting with an investigation of **software structure change**, different parties are involved in sourcing applications: a client company that uses the software artefact to support its business processes and—if a packaged enterprise system is sourced—a software vendor that develops the application. Consequently, software structure change can either occur client-driven or vendor-driven (in on-premises and on-demand settings).

Innovation literature tags these types of software structure change as technology push and business pull innovation (Currie et al. 2004; Davern and Kauffman 2000; Horbach et al. 2012). Vendor-driven software structure change is related to *technological push* innovation (e.g. Carmel and Sawyer 1998; Davern and Kauffman 2000; Kim et al. 2009; Lin and Chen 2012). It describes a situation in which a software vendor performs deep or surface structure transformations of an enterprise system. Software change is governed outside a firm's hierarchy.

Client-driven software structure change is related to *business pull* innovation (e.g. Carmel and Sawyer 1998; Davern and Kauffman 2000; Kim et al. 2009; Lin and Chen 2012). It describes a situation in which deep or surface structure transformation of an enterprise system is either performed by a corporate's IT unit or by a subcontracted external third-party vendor under control of the client. Software innovation is governed within a firm's hierarchy.

Software structure change can either occur client-driven or vendor-driven. Table 2.6 gives the definition of the related technological push and business pull innovation concepts (Carmel and Sawyer 1998; based on Davern and Kauffman 2000; Kim et al. 2009; Lin and Chen 2012).

2.2.5.2 Business Process Structure Change

The second necessary condition of dynamic fit is **business process structure change**. A process is "*a series of actions or steps taken in order to achieve a particular end*" (Oxford Dictionaries 2014b). A specific kind of a process is a business process (Heinrich et al. 2011). Drawing upon institutional theory business processes can be seen as a set of organized and established structures (Berente and Yoo 2012; Sia and Soh 2007; Strong and Volkoff 2010). Business processes link functions for creating a business outcome in a chronological and factual manner (Heinrich et al. 2011). They are sets of logically related tasks, rules, and procedures

Table 2.6 Definition of software structure change

Construct		Definition	Sources
Software structure change	Technology push innovation	Technology push innovation occurs, when software innovation developed by a software vendor is implemented within a firm	Based on Carmel and Sawyer (1998), Davern and Kauffman (2000), Kim et al. (2009), Lin and Chen (2012)
	Business pull innovation	Business pull innovation occurs, when software innovation developed by a client is implemented within its firm	

Table 2.7 Definition of business process change

Construct	Definition	Sources
Business process structure change	Business process structure change occurs, when an organization transforms the structure of its tasks, rules and procedures	Based on Blum (2006), Maltz and Kohli (1996)

that are required for realizing a desired business goal (Blum 2006; Davenport and Short 1990; Maltz and Kohli 1996; Melville et al. 2004).

Like software structures, business process structures are not assumed to be static in nature. Institutional theory argues that most structures are exposed to change over time (Berente and Yoo 2012; Jones and Karsten 2008; Scott 1987). As shown in Table 2.7, business process structure change can be defined as transformation in the structure of tasks, rules, and procedures within an organization (Blum 2006; Maltz and Kohli 1996).

2.2.5.3 Environmental Change

The stimulus for software and business process structure change can be found within the environmental setting of an organization (Scott 1987). Changes in the structures of business processes and software are caused by fluctuations in organization's competitive context (e.g. length of product and service life cycles, customer turnover, and market share), by variations of industry-specific rules or procedures, or by shifts in regulatory frameworks and country-specific laws (e.g. Gemino et al. 2007; Maltz and Kohli 1996; Soh and Sia 2004; Son and Benbasat 2007; Tallon and Pinsonneault 2011; Tiwana and Keil 2009; Wang et al. 2006).

Institutional theory differentiates between imposed and voluntary structural change (Scott 1987; Sia and Soh 2007; Soh and Sia 2004). Imposed change is a result of external forces such as the government or established industry best practice (DiMaggio and Powell 1983; Sia and Soh 2007). It can be distinguished between changes in the firm-specific, the industry-specific, and the country-specific imposed context (Scott 1987; Sia and Soh 2007). The latter one refers to shifts in the socio-

Table 2.8 Definition of environmental change

Construct		Definition	Sources
Macro environmental change	Imposed country context	Changes in the socio-political system, economic configuration, or cultural practices within a country	Based on Maltz and Kohli (1996), Scott (1987), Sia and Soh (2007), Soh and Sia (2004)
	Imposed industry context	Changes in practices specific to firm's industrial sector	
Micro environmental change	Imposed firm context	Changes in structures related to a firm's trading partners	
	Voluntary context	Changes in the idiosyncratic organisational structures	

political system, economic configuration, or cultural practices within a country (Sia and Soh 2007). The introduction of a new national identification number or the inception of SEPA are examples for country-specific changes. Changes in the imposed industry-specific context are transformations in practices specific to firm's industrial sector (Sia and Soh 2007). An example would be a new and more stringent accountability requirement within the pharmaceutical industry (Sia and Soh 2007). Finally, changes in the imposed firm-specific context result from a firm's working relationship with its trading partners (Melville et al. 2004; Scott 1987; Sia and Soh 2007).

While shifts within the imposed setting comes from outside of an organization, changes in the voluntary context result from internal forces (Scott 1987). The voluntary context is idiosyncratic to a particular firm (Sia and Soh 2007). It includes such processes, rules, or procedures that organisations develop as a result of their experience, strategy, and management preferences (Scott 1987; Sia and Soh 2007).

In contrast to shifts in the voluntary context, which are always idiosyncratic to a particular firm (Sia and Soh 2007), imposed context change either occurs within the macro or the micro environment of an organization (Melville et al. 2004). Macro environmental changes are shifts in the imposed country-specific or the imposed industry-specific context (Sia and Soh 2007). These changes impact all companies within a nation or a particular sector. Micro environmental changes are unique to a particular company. They encompass shifts in the imposed firm-specific or the voluntary context of an organization (Scott 1987; Sia and Soh 2007). Table 2.8 gives the definition of macro and micro environmental change.

2.2.5.4 Contextual Factors Related to Software and Business Process Structure Flexibility

Dynamic fit is not merely a result of software and business process structure change (Nöhren et al. 2014). These changes must take place with an adequate timing, magnitude, and direction to be beneficial for a company (Zajac et al. 2000). If a company experiences an environmental change which results in a business process transformation, its enterprise systems needs to change equipollently and isochronally (Berente and Yoo 2012; Maltz and Kohli 1996; Scott 1987; Sia and Soh 2007). Consequently, software structures need to be flexible.

"*Software is flexible if it can be efficiently and rapidly adapted because of a change in business needs*." (Wang et al. 2008b: 438). This flexibility, which can be seen as an inherent and specific characteristic of a sourced software artefact, can be discussed by the help of TCE (Benlian et al. 2009; Winkler and Brown 2014).

TCE is a widely used theoretical lens in IT outsourcing research. It supports researchers by investigating decision, success, and failure of sourcing arrangements (e.g. Benlian et al. 2009; Dibbern et al. 2008; Schwarz et al. 2009). TCE provides a theoretical view on one major characteristic of software sourcing: application specificity (Winkler and Brown 2014). If an application is specific to a particular firm's requirements (high asset specificity), economic benefits from outsourcing decreases (Kern et al. 2002a, b; Winkler and Brown 2014). This application specificity "*is reflected in the degree that specific applications can be customized, integrated, and modularized prior to and in the outsourcing relationship*" (Benlian et al. 2009: 360). It serves as a measure for software's flexibility and its ease of customization (Winkler and Brown 2014). Application specificity was found to impact the adoption of on-demand software (Benlian et al. 2009) as well as the internal governance structure of applications (Winkler and Brown 2014). Its relationship with software alignment has not been studied so far.

In order to meet their idiosyncratic requirements and to realize fit, deep and surface attributes of software are frequently customized by client companies (e.g. Domberger et al. 2000; Maurer et al. 2012; Sarker et al. 2012; Xin and Levina 2008). Enterprise systems can be differentiated according to their customizability. This **software customizability** can be measured in terms of application specificity (Benlian et al. 2009; Winkler and Brown 2014). Customizations are frequently required to embed an enterprise system within an organization (Sia and Soh 2007; Soh and Sia 2004; Strong and Volkoff 2010). Software vendors differentiate themselves from their competitors by the degree to which their applications allow for individual adaptations of deep and surface structures (Slaughter and Levine 2006). Table 2.9 gives the definition of software customizability (Benlian et al. 2009; Wang et al. 2008b; Winkler and Brown 2014).

If an environmental change results in software transformation, organizational business processes must change equipollently and isochronally (Berente and Yoo 2012; Maltz and Kohli 1996; Scott 1987; Sia and Soh 2007). Consequently, not

Table 2.9 Definition of software customizability

Construct	Definition	Sources
Software customizability	Degree to which deep and surface structures of software can be rapidly and extensively customized by a client	Based on Benlian et al. (2009), Wang et al. (2008b), Winkler and Brown (2014)

Table 2.10 Definition of business process adaptability

Construct	Definition	Sources
Business process adaptability	Degree to which structures of business processes can be rapidly and extensively modified and adapted	Based on Lacity et al. (2011), Tatikonda and Montoya-Weiss (2001), Yang and Papazoglou (2000)

only software but also business process structures need to be flexible in nature in order to cope with changing software structures.

"*Process adaptability refers to flexibility (...) to meet emerging circumstances*" (Tatikonda and Montoya-Weiss 2001: 155). This flexibility, which can be seen as an inherent characteristic of organization's business processes, can be discussed by the help of RBV (Barney 1991; Tatikonda and Montoya-Weiss 2001).

RBV differentiates between human, physical, and organizational resources (Barney 1991; Grant 1991). Physical resources are assets like a company's technology, its equipment, and its geographical position (Barney 1991; Nöhren and Heinzl 2012). Human resources include skills and experience of managers and workers within the firm (Barney 1991). Organizational resources are assets such as an internal planning, controlling, and coordinating system as well as processes to fulfil customers' needs (Barney 1991; Nöhren and Heinzl 2012). Consequently, business processes (Heinrich et al. 2011) are organizational resources.

"To remain competitive organizations must be able to move fast and quickly adapt to change. Moreover, they must be able to reconfigure their key business processes as changing market conditions dictate." (Yang and Papazoglou 2000: 43). Organizational business processes differ with respect to their adaptability (Tatikonda and Montoya-Weiss 2001; Yang and Papazoglou 2000). **Business process adaptability** serves as a measure for business process structure flexibility and its ease of adaptation (Tatikonda and Montoya-Weiss 2001; Yang and Papazoglou 2000). Table 2.10 gives the definition of this construct (Lacity et al. 2011; Tatikonda and Montoya-Weiss 2001; Yang and Papazoglou 2000).

2.2.5.5 Summary

Software-business process fit is non-deterministic and dynamic in nature (Sabherwal et al. 2001; Vessey and Ward 2013). It expresses the coevolution between software and business process structure over time (Nöhren et al. 2014; Sabherwal et al. 2001; Zajac et al. 2000). This section discusses components of the process by which this coevolution in terms of dynamic fit is generated. Drawing upon institutional theory in combination with RIT, software structure change and business process structure change were identified as necessary conditions for dynamic fit. The latter one is defined as transformation in the structure of tasks, rules, and procedures within an organization (Blum 2006; Maltz and Kohli 1996). Software structure change can occur client-driven and vendor-driven. Vendor-driven software structure change refers to technology push innovation and expresses the implementation of a software innovation developed by a software vendor within a firm (Carmel and Sawyer 1998; based on Davern and Kauffman 2000; Kim et al. 2009; Lin and Chen 2012). Client-driven software structure change refers to business pull innovation and describes the implementation of a software innovation, which is developed by a client (Carmel and Sawyer 1998; based on Davern and Kauffman 2000; Kim et al. 2009; Lin and Chen 2012).

Both, business process structure and software structure change are results of shifts in environmental conditions (Scott 1987). These shifts are either firm-specific and result from internal or external environment (micro environmental change) or impact all companies within a country or industrial sector (macro environmental change) (Maltz and Kohli 1996; Scott 1987; Sia and Soh 2007; Soh and Sia 2004).

Due to the fact that dynamic fit is not merely a result of change in software and business process structure but also that these changes take place with an adequate timing, magnitude, and direction (Nöhren et al. 2014; Zajac et al. 2000), two contextual factors related to flexibility of software and business process structures were identified. Software customizability was defined as the degree to which deep and surface structures of software can be rapidly and extensively customized by a client (Benlian et al. 2009; Wang et al. 2008b; Winkler and Brown 2014). Business process adaptability describes the degree to which structures of business processes can be rapidly and extensively modified and adapted (Lacity et al. 2011; Tatikonda and Montoya-Weiss 2001; Yang and Papazoglou 2000).

2.3 Summary

The presented study aims to contribute to the discussion of how software sourcing modes impact software sourcing value in terms of alignment and performance and how these outcomes are interrelated with each other. No strong and precise definition of in-house, on-premises, and on-demand software was found in previous researches. Drawing upon the representational view of IT (Sia and Soh 2007; Strong and Volkoff 2010; Wand and Weber 1990), software sourcing modes were defined based on client company's ownership of physical, deep, and surface structures. In an in-house setting, physical, deep, and surface structures of the software artefact are held within a firm's hierarchy. In contrast to this, the ownership on deep and surface structures of packaged applications is held by a software vendor. On-premises and on-demand software differ with respect to their physical structures in terms of their deployment. Whereas on-premises applications are installed on a firm's own IT infrastructure (internal physical structure ownership), on-demand applications are hosted at a software vendor (external physical structure ownership) and are accessed via Internet. In a consequence, the extent of ownership on physical, deep, and surface structures decreases from in-house through on-premises to on-demand sourcing.

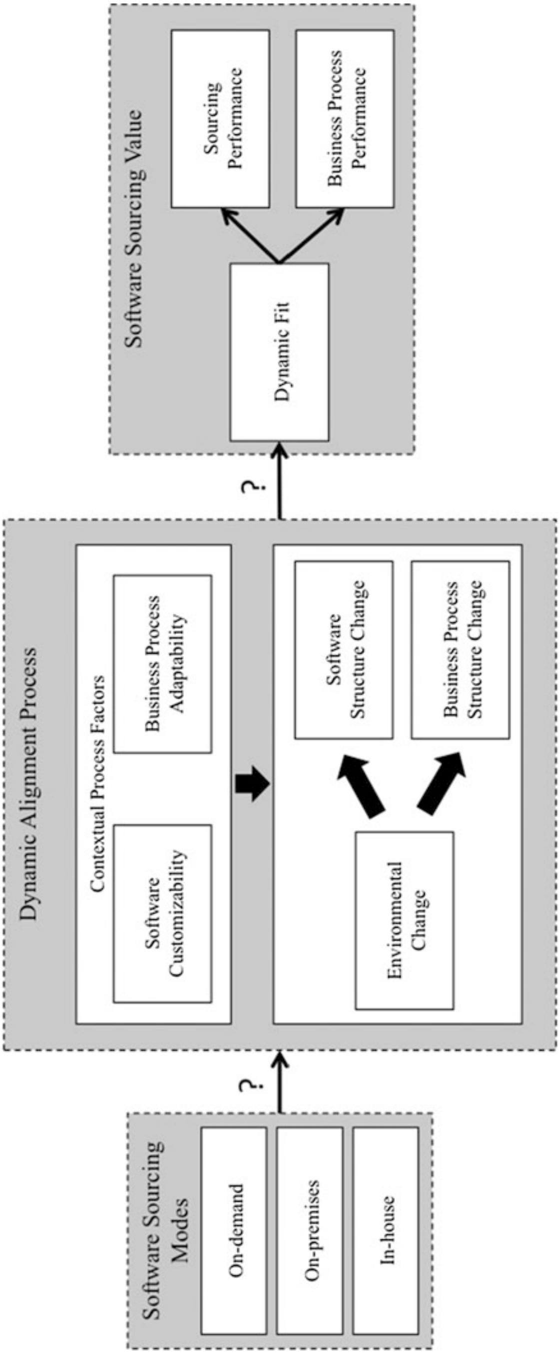

Fig. 2.14 Summary of this chapter

Drawing upon a systematic review of literature, three outcome concepts are encompassed in this study. First, software-business process fit is included. Taking a representational view of IT perspective, misfits occur as "*(...) the result of differences between the structures embedded in the package and those embedded in the organization*" (Soh and Sia 2004: 375). In previous research, three perspectives of fit between software and business process structures emerged: the notion of Sia and Soh (2007), the notion of Strong and Volkoff (2010), and the notion of Nöhren et al. (2014). By combining the concept of Strong and Volkoff (2010) with the concept of Nöhren et al. (2014), a new perspective on dynamic fit between software structure and business process structure emerged.

The remaining two outcome concepts identified by the literature review refer to business process and sourcing performance. Business process performance captures software's influence within the business units (Grant 2003; Shang and Seddon 2002). Sourcing performance determines enterprise system's effect on the IT function in terms of its contribution to strategic, economic, and technological benefits (e.g. Goo et al. 2008; Grover et al. 1996; Lacity and Willcocks 2001; Lee and Kim 1999; Lee et al. 2004; Saunders et al. 1997). Based on previous findings it can be concluded that a software artefact that fits with the requirements of a firm impacts performance outcomes (Chan and Reich 2007; DeLone and McLean 1992; Soh and Markus 1995; Strong and Volkoff 2010). Therefore, dynamic fit is attributed to be an intermediary outcome factor impacting sourcing and business process performance.

It was found that software alignment is rather a process than an end in itself (Chan and Reich 2007; Sabherwal et al. 2001; Vessey and Ward 2013). Based on institutional theory in conjunction with representational view of IT, dynamic fit results from software structure and business process structure change (Scott 1987; Sia and Soh 2007; Wand and Weber 1990). These changes are results from shifting environmental conditions (Berente and Yoo 2012; Maltz and Kohli 1996; Scott 1987). This dynamic alignment process, with dynamic fit as outcome, is impacted by flexibility of software structures and flexibility of business process structures as contextual factors (Sia and Soh 2007; Tatikonda and Montoya-Weiss 2001; Wang et al. 2008b; Winkler and Brown 2014; Yang and Tate 2012).

No study was found that compares in-house, on-premises, and on-demand software with each other in terms of their alignment and performance. Consequently, the role of software sourcing modes remains a "black box". The presented study aims to open this "black box" by studying how in-house, on-premises, and on-demand software is related to the dynamic alignment process, dynamic fit, business process performance, and sourcing performance.

The conceptualization of this chapter is summarized in Fig. 2.14. It forms the basis for the development of a preliminary research model presented in Chap. 3. By combining process logic with variance logic (Sabherwal and Robey 1995), the dynamic alignment process will be transferred into stable and testable clusters grouped into gestalts and non-gestalts (Lee et al. 2004; Venkatraman 1989). The relationship between these patterns and software sourcing modes as well as their influence on software sourcing value will be discussed.

http://www.springer.com/978-3-319-23924-8

Enterprise Software Sourcing Performance
The Impact Logic of On-Demand, On-Premises, and
In-House Software on Dynamic Fit and Process-Level
Performance Outcomes in Client Organizations
Nöhren, M.
2016, XXI, 223 p., Hardcover
ISBN: 978-3-319-23924-8