Tailor the Value-Based Software Quality Achievement Process to Project Business Cases

Liguo Huang[1], Hao Hu[2], Jidong Ge[2], Barry Boehm[1], and Jian Lü[2]

[1] Computer Science Department, University of Southern California,
Los Angeles, CA 90089-0781, USA
{liguohua, boehm}@sunset.usc.edu
[2] State Key Lab for Novel Software Tech., Institute of Computer Software,
Nanjing University, Nanjing, 210093, China
{myou, gjd, lj}@ics.nju.edu.cn

Abstract. This paper proposes a risk-based process strategy decision-making approach. To improve the flexibility in applying the Value-Based Software Quality Achievement (VBSQA) process framework, we embed the risk-based process strategy decision-making approach into the VBSQA process framework. It facilitates project managers to tailor the VBSQA process framework to different project business cases (schedule-driven, product-driven, and market trend-driven). A real world ERP (*E*nterprise *R*esource *P*lanning) software project (DIMS[1]) in China is used as an example to illustrate different process strategies generated from process tailoring.

1 Introduction

1.1 VBSQA Process Framework

Value-Based Software Quality Achievement (VBSQA) process framework [1] is generated from the WinWin Spiral Model [2] and the theories of value-based software engineering [3]. It provides a general guideline to generate process instances in order to achieve stakeholder WinWin-balanced software quality requirements based on risk-driven concurrency. Instead of using one-size-fits-all metrics to measure software quality achievement, VBSQA process framework enables its users to elicit success-critical stakeholders' value propositions (i.e., prioritization, expected & desired values) with respect to quality (Q-) attributes. It also helps identify and resolve their value conflicts on Q-attributes through risk analysis, technology/architecture evaluation and milestone reviews. Note that we also consider schedule and cost as Q-attributes in software projects. Furthermore, the framework guides us to use real earned value to monitor and control the progress toward achieving the Q-attribute requirements. The top-level steps and anchor point stakeholder commitment milestones (bolded) of VBSQA process framework [1] are listed in Table 1.

[1] This DIMS project is anonymous for the sake of commercial confidentiality.

Q. Wang et al. (Eds.): SPW/ProSim 2006, LNCS 3966, pp. 56 – 63, 2006.

Table 1. The top-level steps of VBSQA process framework

1.	Identify top-level mission objectives and stages – including quality (Q-) objectives
2.	Perform project cost/benefit analysis – Estimate project budget – Develop results chain to identify success-critical stakeholders and their top-level value propositions
3.	Stakeholders negotiate mutually satisfactory (Win-Win) quality (and other) goals and relevant mission scenarios.
4.	Concurrently engineer top-level Q-attribute and other requirements and solution tradeoff spaces.
5.	Identify top-level Q-risks, execute risk-mitigation spirals.
6.	Develop system top-level design and initial Feasibility Rationale Description (FRD).
7.	**Hold Life Cycle Objective (LCO) Review** – **Pass: go to 8. Fail: go to 4.**
8.	Concurrently engineer detailed Q-attribute and other requirements and solutions; resolve risks.
9.	Develop system detailed design and detailed Feasibility Rationale Description (FRD).
10.	**Hold Life Cycle Architecture (LCA) Review** – **Pass: go to 11. Fail: go to 8.**
11.	Construct, test, and deploy system – Use the mission scenarios and Q-attribute requirement levels as progress metrics and test cases – **Core Capability Demo (CCD)** – Monitor progress and change requests; perform corrective actions
12.	**Initial Operational Capability (IOC) Readiness Review**

1.2 Implications of Applying the VBSQA Process to ERP Software Development in China

VBSQA process framework covers all phases and milestones through the entire software development life cycle in the WinWin Spiral model [2]. It also includes various software development activities to incorporate the value-based consideration. For most ERP (*E*nterprise *R*esource *P*lanning) solution providers in China, different software quality assessment criteria are set based on different business cases [5] and different process strategies shall be selected to meet them. Three process strategies (schedule-driven, product-driven and market-trend driven) can be selectively applied in the ERP software development based on different business cases. Therefore a flexible process generation platform is expected to enable the trim or addition of the steps/milestones/activities in VBSQA process framework.

2 Process Strategy Decision-Making in VBSQA Process

In general, schedule-driven processes are lightweight processes that employ short iterative cycles while product-driven processes employ longer iterative cycles. Our risk-based process decision-making approach, summarized in Fig. 1, uses the project business case and risk analysis to tailor the VBSQA process into an overall software development strategy [4]. Embedding the risk-based process decision-making approach into VBSQA process framework provides a feasible solution to a flexible process generation based on project business cases. This approach relies heavily on project key stakeholder identification, project business case analysis and the collaboration of the core development team with other project stakeholders. Thus we insert the process decision-making point after Step 3 (stakeholders negotiate quality and other goals) in the current VBSQA process framework described in Table 1.

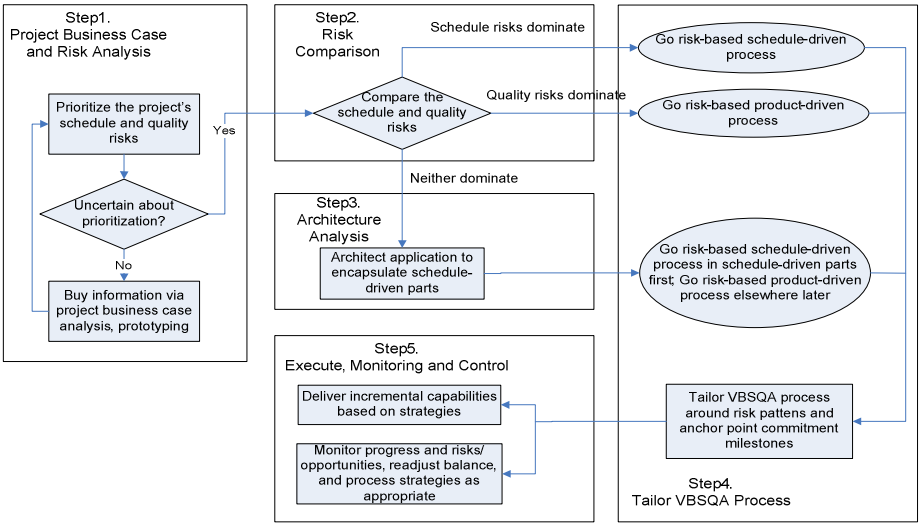

Fig. 1. Summary of risk-based process decision-making approach

Business case analysis aims to elicit success-critical stakeholders' value propositions. Risk analysis aims to identify and mitigate risks particularly associated with project schedule and software quality achievement. The results of risk analysis can be used to answer such questions as "How much software quality investment is enough?" by balancing the risk of investing *too little* on software quality with the risk of investing *too much*. Examples of such questions related to software quality achievement are "How much prototyping is enough?", "How much review is enough?", and "How much testing is enough?" As another aspect of quality achievement, we extend the approach to also consider the question "How much architecting and planning is enough?" Risk analysis is closely related to business case analysis in that project risks are prioritized based on the business case analysis by emphasizing the high-priority stakeholder value.

If schedule risks dominate quality risks, risk-based schedule-driven process is applied. If quality-risks dominate schedule risks, risk-based product-driven process is applied. If neither dominates, then architect the application to encapsulate the schedule-driven parts which applies the risk-based schedule-driven process and go risk-based product-driven process elsewhere. Based on this approach, we can tailor the VBSQA process framework and establish an overall project strategy by integrating individual risk mitigation plans [4].

Since no decision is perfect for all time, as indicated in Step 5 in Fig. 1, project management team needs to continuously monitor and control the performance of the selected process in order to adapt to changes in the project business case. In this way, we can always monitor and control the opportunity for realizing stakeholders' value.

3 Tailor VBSQA Process to Different Business Cases

Using the process decision-making approach embedded in the VBSQA process framework, we are able to tailor the VBSQA process to different project business cases when generating a process instance for a software project. When tailoring the process, we may skip some process steps/milestones, relax the deliverables/outputs of a particular process step/milestone, select a particular process activity or decide the participants of a process activity.

3.1 Characteristics of Three Example Business Cases

We first determine whether the project is dominated by schedule risks or quality risks before process tailoring. Table 2 compares the different characteristics of three typical business cases in ERP software projects. Then we use a real-world ERP software system, a Documents and Images Management System (DIMS) built by Neusoft, as an example to illustrate how to use the risk-based process decision-making approach to tailor the VBSQA process to three different business cases. Four success-critical stakeholder classes were identified in DIMS project, including the System Acquirer, DB Administrators, Software Maintainers and Developers.

Table 2. Characteristics of Three Example Business Cases in ERP Software Development

Business Case	Schedule-Driven	Market Trend-Driven	Product-Driven
Primary Objective	Rapid value by adding small extra functionalities	Rapid Market Share Occupation	Version upgrade with Q-attribute achievement: reliability, availability, performance, evolvability, etc.
Quality Risks	Low	Medium	High; major business losses
Schedule Risks	High; major business losses	High; market share loss	Low
Stakeholders	Single collocated representatives	Many success-critical stakeholders	Multiple success-critical stakeholders with various Q-attribute requirements
Requirements	1) A few specific and stable requirements; 2) Mostly functional	1) Goals generally known (e.g., platform changes); 2) Detailed requirements often vague, volatile and emergent; 3) Functional and non-functional [6]	1) Critical and conflicting Q-attribute requirements from various stakeholders; 2) Most requirements relatively stable; others volatile, emergent 3) Functional and nonfunctional;
Architecture	1) Extend from existing system architecture 2) Little architecting effort 3) Stakeholder high confidence	1) Brand new architecture; 2) Most architecting effort; 3) Stakeholder low confidence	1) Evolve based on existing product-line architecture 2) High confidence in some parts; low confidence in others
Refactoring	Inexpensive with skilled people	More expensive with mix of people skills	Very expensive, with mix of people skills

3.2 Tailor VBSQA Process to Schedule-Driven Business Case

Schedule-driven business case applies when rapidly accommodating a few minor product upgrading requirements from one or two departments within an organization. The examples of such requirements can be adding, deleting, updating certain attributes in the current DIMS database schema. Those functionalities are usually needed urgently. Delivering the functionalities on time becomes the stakeholders' highest-priority value proposition. Thus, we need to prioritize the process steps/activities and

tailor the VBSQA process framework to only retain the most effective process steps/milestones/activities. In this case, system users are willing to tolerate some quality degradation and delay the Q-attribute requirements until the system operation.

Based on the schedule-driven business case in Table 2, the added functionalities are extended from the existing system architecture and stakeholders are more confident in the existing architecture. There is no need to propose or review several feasible architectural options. Requirements are specific enough to skip the high-level design and to proceed directly to the detailed design stage. In this case, process steps in Life Cycle Objective (LCO) stage are less effective than those in the Life Cycle Architecture (LCA) stage in VBSQA process framework. For the same reason, we may also skip the intermediate milestone Core Capability Demo (CCD) and proceed to Initial Operational Capability (IOC) Readiness Review. Since the quality risks are relatively low and developers only need to extend from the existing system architecture, *Selected Architectural Internal Review* within the developer team is performed in LCA stage instead of onsite *External Review* with the participation of all stakeholders in order to meet the delivery deadline. Fig. 2 shows an example of schedule-driven process strategy for DIMS project.

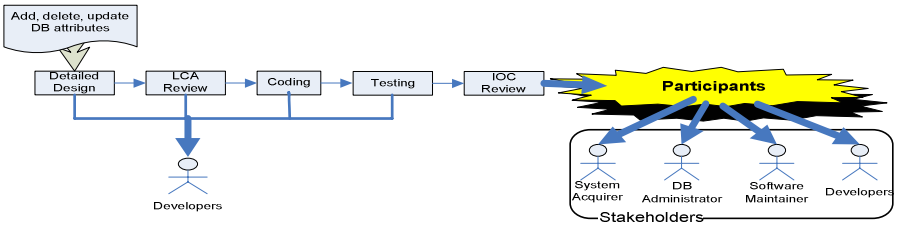

Fig. 2. An example of schedule-driven process strategy for DIMS project

3.3 Tailor VBSQA Process to Product-Driven Business Case

Product-driven business case applies when accommodating a system upgrading request to a higher version after aggregating common upgrading requirements from various departments. In this case, quality of the upgraded product is the process driver rather than meeting a delivery deadline. Quality risks are dominant compared with schedule risks as shown in Table 2. The requirements are relatively stable. Since the requirements are aggregated from various project stakeholders, the Q-attribute requirements may conflict with one another. An example is the DIMS version upgrade from 6.0 to 7.0. Functional requirements and their associated Q-attribute requirements were prioritized through stakeholder WinWin negotiation in Table 3.

Multiple project increments can be proposed based on the priorities of requirements. A process instance is generated for each increment. Fig. 3 shows an example of product-driven process strategy in DIMS upgrade project. R1, R2, R3 and R4 are grouped into the first increment due to their higher priorities and cohesion. R5 and R6 are grouped into the second increment. In product-driven business case, process instances of multiple increments can proceed concurrently since the functional and

Table 3. Prioritized requirements in DIMS upgrade from 6.0 to 7.0

Requirements	Description	Category	Priority
R1	Data migration from old DB platform to upgraded DB platform	Functional	High
R2	Data migration shall be completed within 1 day and within the storage space	Quality (Performance)	High
R3	Accommodate different DB platforms and schema in data migration	Quality (Evolvability)	Medium
R4	Add a printing function in DIMS system	Functional	High
R5	Build a unified log in user interface for different DIMS subsystems	Functional	Medium
R6	Improve search response time from 2 seconds to 0.5 seconds	Quality (Performance)	Medium

Q-attribute requirements are relatively stable. In each increment, process strategy shall place emphasis on involving stakeholders in identifying and resolving conflicting Q-attributes, concurrently identifying and mitigating Q-risks with architecture/technology evaluation and milestone reviews. Thus, its iteration cycle is longer than schedule-driven process in order to address the Q-risks and maintain the product-line architecture.

LCO/LCA reviews and CCD are all necessary to identify and mitigate Q-risks in each increment. It is also important to involve all success-critical stakeholders in the prototype evaluation and each milestone review (i.e., LCO, LCA, CCD, IOC). Therefore, performing onsite *External Prototype Evaluation*, *Architecture Options External Review* and *Selected Architecture External Review* with the participation of the System Acquirer, DB Administrators, Software Maintainers and Developers, is more effective than their internal counterparts within developer team.

3.4 Tailor VBSQA Process to Market Trend-Driven Business Cases

Market trend-driven business case applies when the upgrade of the product is driven by the market trend or competing companies' products, such as a change from Client/Server architecture to web-based architecture in the DIMS. In this case, providing superior capabilities to capture greater market share as early as possible is the key process driver.

The priorities of schedule risks and quality risks are comparable for market trend-driven business case as shown in Table 2. Therefore, the process strategy for market trend-driven business case is a mixture of the schedule-driven and product-driven process strategies. It is similar to schedule-driven process strategy in that it maintains the short iteration cycle in the first project increment to meet the product delivery deadline for capturing the market share early. However, since stakeholders are less confident in the web-based architecture, it is different from schedule-driven process strategy in that stakeholders should be closely involved in the prototype evaluation and each milestone review (LCO, LCA, CCD, IOC) as shown in Fig. 4.

It is similar to product-driven process strategy in that it emphasizes stakeholder involvement and multiple project increments can be proposed based on stakeholders' priorities of functional and Q-attribute requirements. However, it is different from product-driven process strategy in that only the top-priority capabilities can be

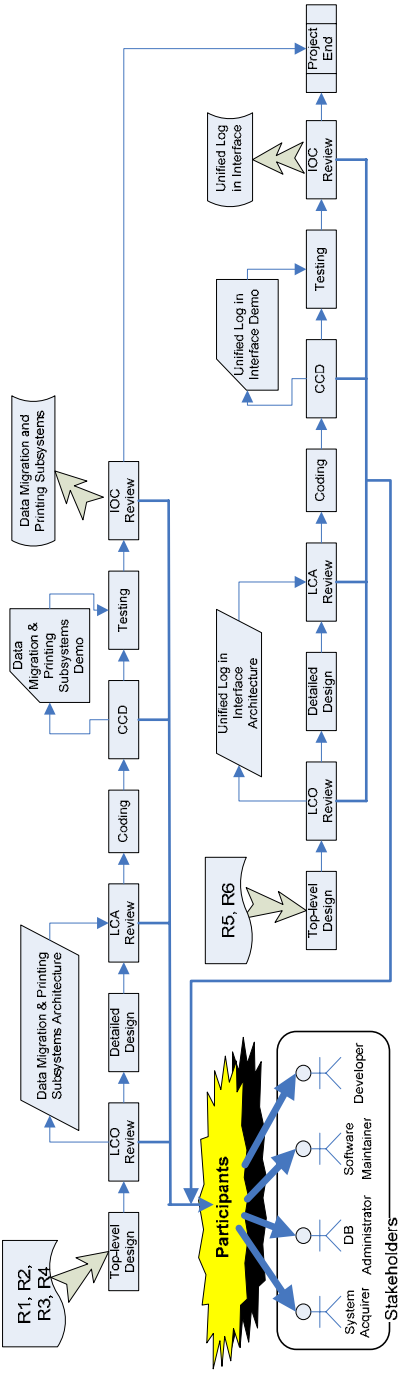

Fig. 3. An example of product-driven process strategy for DIMS version upgrade

Fig. 4. An example of market trend-driven process strategy for changing from Client/Server-based DIMS to web-based DIMS

accommodated in the first increment (see Fig. 4) based on the Schedule/Cost/Quality as Independent Variable (SCQAIV) process strategy [7]. Stakeholders are usually willing to tolerate some quality (e.g., performance, evolvability) degradation at the initial trial of the system. In addition, the process strategy in the following increments heavily depends on the market feedback of the product delivered in the first increment. Thus, there is a gap between each increment to wait for the market feedback. As the operation of new platform becomes stable with sufficient market feedback, product-driven process strategy can be applied in the following increments.

4 Conclusion and Future Work

The risk-based process strategy decision-making approach embedded in VBSQA process framework enables us to tailor the process to various project business cases. It improves the flexibility in applying the process framework. Business case and risk analyses are critical success factors in selecting an appropriate process strategy.

We are investigating the interactive tool support for project managers to tailor the VBSQA process framework and generate an appropriate process instance for a specific project business case.

References

1. Huang, L.: A Value-Based Process for Achieving Software Dependability", Proceedings of International Software Process Workshop (2005), Beijing, China. LNCS, Springer Verlag
2. Boehm, B., Hansenzz, W.: Understanding the Spiral Model as a Tool for Evolutionary Acquisition", CrossTalk, May, (2001)
3. Boehm, B., Jain, A.: An Initial Theory of VBSE, in A. Aurum, S. Biffl, B. Boehm, H. Erdogmus, and P. Gruenbacher, Value-Based Software Engineering, Springer Verlag (2005)
4. Boehm, B., Turner, R.: Balancing Agility and Discipline, Addison Wesley, (2004)
5. Reifer, D.: Making the Software Business Case, Addison Wesley, (2002)
6. Chung, L., Nixon, B., Yu, E., Mylopoulos; J.: Non-Functional Requirements in Software Engineering, Kluwer, (1999)
7. Boehm, B., Port, D., Huang, L., and Brown, W.: Using the Spiral Model and MBASE to Generate New Acquisition Process Models: SAIV, CAIV, and SCQAIV, CrossTalk, vol. 15, no. 1, January, (2002), pp. 20-25.