Tailoring Software Development

Thomas W. Ferratt
MIS, OM, & DSC
School of Business Administration
University of Dayton
300 College Park
Dayton, Ohio, USA 45469
937.229.2938
ferratt@udayton.edu

Bin Mai
Department of MIS
College of Business
Bowie State University
14000 Jericho Park Road
Bowie, Maryland, USA 20715
301.860.4019
bmai@bowiestate.edu

ABSTRACT
In this research-in-progress we review existing studies and theories that focus on tailoring software development. Our purpose is to develop a more complete model of the tailoring process than is found in the current literature. In our theory, "tailoring" refers to the activities of selecting a set of standardized processes for a software development project prior to the start of the project, modifying those selected processes for the specific project as the development team begins its work, and monitoring and adjusting the processes and project goals and constraints over the course of the project. We present a model that allows us to understand what is involved in tailoring software development at key points in time relative to a specific software development project. We present the model as currently a work-in-progress and illustrate its use by presenting a few hypotheses based on relationships specified in the model and supporting literature.

Categories and Subject Descriptors
D.2.9 [Software Engineering]: Management, K.6.1 [Management of Computing and Information Systems]: Project and People Management – *Systems development*, K.6.3 [Management of Computing and Information Systems]: Software Management – *Software development, Software process*

General Terms
Management, Theory

Keywords
Managing software projects, configurable development processes, process line, modifying project goals and constraints, adjusting project budget, schedule, and quality, tailoring by stage, factors affecting tailoring, aligning software processes with strategy

SIGMIS-CPR'10, May 20–22, 2010, Vancouver, BC, Canada.

1. INTRODUCTION
Software development is based on a set of processes that are typically conducted by a development team working on a project to provide a software product with specific required characteristics within time and budget constraints. Usually, software development processes that are predictable are more desirable (Forrester 2006, Nidumolu and Knotts 1998, Paulk et al. 1993). Firms that can provide accurate estimates of the cost and time needed to provide software with specific capabilities have an overall software development process that is highly predictable. Xu and Ramesh (2007) refer to well-managed, predictable, repeatable, pre-determined software development processes as methodical software development. However, a number of assumptions underlying methodical software development may not be met in reality, as we will discuss later. As a result, following pre-determined processes, i.e., processes selected at the outset of a project, may not lead to predictable results, i.e., producing software that meets user requirements within resource constraints. Attempts to avoid such an undesirable situation, i.e., unpredictable results, could involve modifying not only the processes the development team uses to produce the software but also the goals and constraints of the software development project, including the required product characteristics or the resources allocated to the project. Tailoring software development by modifying development processes or project goals and constraints is the focus of this paper.

In this paper, we use the term "tailoring" to refer specifically to the procedure of determining the optimal set of standardized software development processes for a software development project, modifying the selected software development processes according to a set of criteria, and monitoring and adjusting the processes and project goals and constraints during the software development period. This definition of "tailoring" shares some important common ground with the software tailoring literature, yet focuses on a significant aspect of software tailoring that is not sufficiently addressed in existing literature.

Germonprez et al. (2007) provided a good synthesis of the tailoring aspect of information technology. Similar to their work, our investigation into software development tailoring "incorporates aspects of the specification of IS design theories of Gregor and Jones (2007) and Walls et al. (1992), who state that 'the purpose of design theory is to support the achievement of goals'" (p. 353). Also, similar to their theorization of "tailorable technologies", our approach to the software development tailoring

"need[s] to consider the broader social context in which technologies may be used" (p. 353). However, our focus in this paper bears a fundamentally significant difference from Germonprez et al.'s (2007) work. Whereas their work theorizes how to design a technology that would allow the eventual users to tailor the pieces of the technology, and only hints at tailoring in the process of development, our theory focuses on the development process, and investigates how a development team and those responsible for sponsoring and managing a software development project (1) select and modify standardized software development processes to develop a system at the outset of a project and (2) monitor and adjust the processes and project goals and constraints throughout the project. This focus represents the unique contribution of our theory. We specifically address the stages and levels at which tailoring occurs and factors affecting tailoring decisions.

Determining the processes to select at the outset of a project, making modifications to selected processes, or modifying other elements of software development when the assumptions underlying methodical software development are not met all may be described as three different aspects of tailoring software development. A number of works have contributed to our understanding of this important phenomenon (Xu and Ramesh 2007, Slaughter et al. 2006, Forrester 2006, Adler 2005, Cameron 2002, Nidumolu and Knotts 1998, Paulk et al. 1993). However, the theoretical models in these individual works do not make it easy to understand tailoring options at key points in time, from prior to the inception of a development project to the implementation of specific processes during a project as well as throughout a project as progress is monitored. Our objective is to provide such a model.

In this research-in-process we identify a few key studies and theories of tailoring software development that serve as the basis for a more complete review and discussion. We present and describe the current version of our model. With further development it will serve as the foundation for presenting hypotheses in a more complete paper. Finally, we illustrate the value of the framework by briefly presenting a limited set of hypotheses.

2. THEORIES OF TAILORING SOFTWARE DEVELOPMENT

In our theory, software development tailoring takes place at three levels: selecting the optimal set of software development processes at the outset of a project by those responsible for sponsoring and managing a software development project; modifying the selected processes by those conducting the project at the outset of the project; and monitoring the software development project and making adjustments to processes and project goals and constraints along the way. A number of different approaches to software development exist (e.g., see Iivari et al. 2001). Different approaches represent differences in the set of interrelated tasks or processes that guide the behavior of developers. The selection of a set of processes or process line (Forrester 2006) for a specific software development project at the outset of the project is one form of tailoring software development. Our theory in a way consolidates the various existing tailoring theories, and presents a comprehensive model for how software development tailoring is achieved. Moreover, our model details the factors we believe are significant in affecting the tailoring decision and illustrates the interrelations among those factors.

We next describe the details of our theory.

2.1 Tailoring by Selecting Processes

Xu and Ramesh (2007) note that we have a limited understanding of the factors affecting tailoring; nevertheless, Slaughter et al. (2006) present a theory of aligning software processes with strategy. Alignment refers to the degree to which business unit strategy, volume of customers, level of product customization, and level of process customization are consistent, as specified below for highly customized, intermediate, and highly standardized alignments:

- Highly Customized Alignment: business unit strategy (differentiation rather than cost leadership), volume of customers (small), level of product customization (high), and level of process customization (high)
- Intermediate Alignment: business unit strategy (intermediate to cost leadership and differentiation), volume of customers (medium), level of product customization (medium), and level of process customization (medium)
- Highly Standardized Alignment: business unit strategy (cost leadership rather than differentiation), volume of customers (large), level of product customization (low), and level of process customization (low)

Slaughter et al. (2006) study firms that provide their customers with an Internet application/service bundle. Product customization refers to the creation of unique features to meet each customer's needs. Process customization refers to tailoring or adapting the development process for each application or customer rather than using a standard development process. Their findings show that "many of the firms match their software process choices to product characteristics, customer volume, and business unit strategies" (Slaughter et al. 2006, p. 891). Thus, this theory of alignment identifies factors that affect the tailoring of software development through the selection of standard or customized processes.

One concern regarding the three alignment configurations is that it does not appear possible to have efficient and effective mass customization (high volume of customers with high level of product customization). See Squire et al. (2006) for full and partial customization vs. standardization. Cameron (2002) views mass customization, in which product and process are both customized to the customer's needs, as desirable and proposes achieving it through the use of configurable work product descriptions (WPDs).

WPDs describe tangible things produced. They describe what the work product is, why and when it is needed, and how it is produced. Selecting and sequencing WPDs are major elements in tailoring or configuring the software development process. This tailoring occurs at the outset of the project. According to Cameron (2002), factors affecting this tailoring include project scope, client responsibility, experience of the team, quality of input materials, and areas of project risk.

Cameron's (2002) implicit theory is that tailoring contributes to achieving efficiency and effectiveness in software development. Similarly, Slaughter et al.'s (2006) implicit theory is that

alignment contributes to achieving efficiency and effectiveness in software development. Efficiency refers to completing the project on time and within budget. Effectiveness refers to building a software product with specified characteristics. Slaughter et al.'s (2006) theory is more explicit than Cameron's, but Cameron (2002) views mass customization as feasible whereas it does not appear to be feasible in Slaughter et al. (2006).

2.2 Tailoring Selected Processes

Besides identifying an overarching set of factors affecting the selection of software development processes as a form of tailoring at the outset of a project (Slaughter et al. 2006, Cameron 2002), we suggest that tailoring of the selected processes occurs as they are encountered and performed by those on the development team. In Cameron's (2002) description of the development process each project has a set of selected processes with WPDs. The specification of that content may also lead to a more detailed tailoring of specific WPD templates (per email from John Cameron, Oct 29, 2008).

2.3 Modifying Various Elements of Software Development

Tailoring also occurs through modifications emerging from the results of monitoring a project and providing feedback on the development process. Xu and Ramesh (2007) report that a software development project includes constant social negotiation and consensus building. Not only are processes modified, but the project and its context are also modified as a result. These modifications and the implementation of tailored processes result in re-specified product requirements, revised resource contraints, and reassessed project risks.

Monitoring the project and providing feedback or constant social negotiation and consensus building would not be necessary if the assumptions underlying a methodical or deterministic view of software development were met. Modifications to goals, resources, or processes occur during a project because one or more of the following assumptions are violated:

- Resources and product goals/characteristics (i.e., desired system capabilities and quality) are accepted/agreed to by all participants
- Processes selected are those needed to achieve goals
  - Desired business objectives determine the desired user/client support needed which, in turn, determine the desired system capabilities and quality which, in turn, determine the processes and resources needed for the project, all of which are known at the start of the project
- Resources allocated are resources needed to conduct processes needed to achieve goals
  - Resources allocated at the start of the project are used to conduct processes selected at the start of the project; those processes lead to achieving the desired system capabilities and quality goals, which lead to achieving the desired direct user effect and, in turn, to achieving desired business objectives
- Processes are conducted effectively by participants, including developers, managers, and users/clients
  - No variations in processes occur once they are selected. There are no differences in developers (e.g., skills, motivation), managers (e.g., managerial skills, approach to managing and controlling, and motivation), organizational practices that could influence developers and managers (e.g., human resource practices), or clients (e.g., their participation or motivation)
- Outcomes achieved are identical to goals
  - Variations in goals achieved (i.e., outcomes) are due to
    - variation in resources allocated at the start of the project
    - variation in processes selected at the start of the project
    - variation in agreed upon goals at the start of the project
- The forces leading to identification and selection of the project (with its desired goals, selected processes, and allocated resources) remain constant

3. MODEL FOR UNDERSTANDING THE TAILORING OF SOFTWARE DEVELOPMENT

Our theory dictates the three levels that tailoring occurs during software development. As a consequence, this model represents a continuous and cyclic procedure of software system development which includes three distinct phases.

In our model, there is a system user who wants a software system to perform certain functions. There is also a software systems provider who has the resources to produce the system for the system user. Together they design the target product. Subsequently, a development team is formed. It designs the project for system development and tailors and implements the selected processes. The system user, system provider, and development team also monitor the project performance and provide feedback. They may also modify the product design and project accordingly.

We model software development in three stages, represented by the three figures below. In the figures, the dotted boxes represent different entities. The solid boxes inside the entities represent the properties of the entities. The annotated dotted arrows represent the action relationships between the entities, and the solid arrows represent the effect relationships between entity property and entity action. The solid arrows are the focus of our interest in hypothesis development.

3.1 Stage 1: Product Design and Process Selection

In this stage, system user and system provider negotiate together, and based on their respective business strategy and operation characteristics, design the software product specification and resource parameters. They also select and align the software development processes. This stage captures two of the characteristics identified by Germonprez et al. (2007) as definitional for a tailorable system: a dual design perspective and user engagement (p. 355). This process selection and alignment will be based on the product characteristics as well as the system user and provider properties, as represented by the solid arrows in Figure 1.

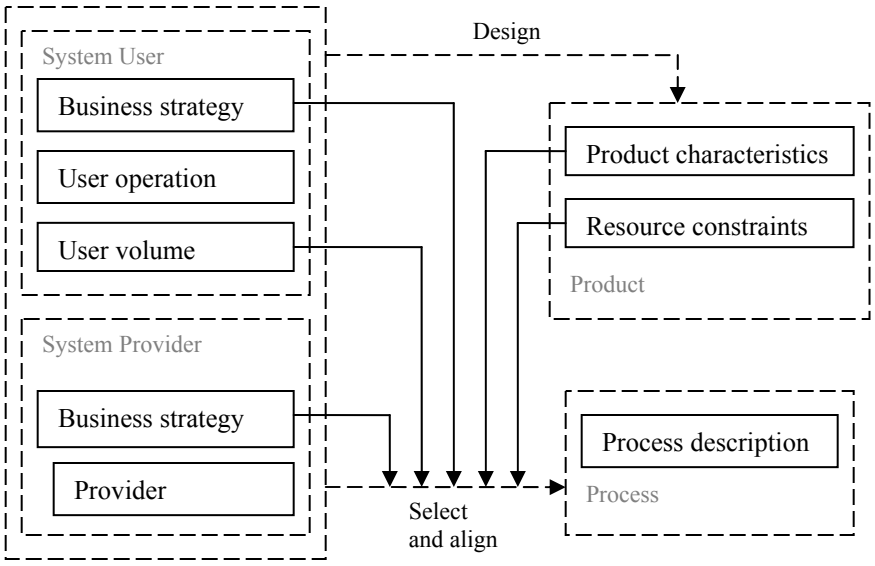

Figure 1: Product design and process selection

Slaughter et al.'s (2006) study revealed several levels for software development process alignment, which showed that business strategy, user volume, and product characteristics are related to process alignment. Based on their alignment configuration, we have the following three hypotheses:

- If the business unit strategy is to be a cost leader rather than differentiator, the software process selection and alignment will be more standardized.
- If the user volume is low, the software process selection and alignment will be less standardized.
- If the product standardization is high, the software process selection and alignment will be more standardized.

Also, Resource-Based Theory (Barney 1991) illustrates the impact of an organization's resources (e.g., time, finance, and expertise) on the firm's internal capabilities. That theory has been applied in various IT/IS contexts, such as IT adoption (e.g., Caldeira and Ward 2003) and IS implementation (e.g., Thong 2000). In our research context, we propose the following hypothesis:

- If the resources for product development are limited, software process selection and alignment will be more standardized.

3.2 Stage 2: Project Design and Selected Process Tailoring and Implementation

In this stage, a development team is formed to produce the software system (i.e., product) based on the parameters designed in Stage 1. The development team may be composed of members from the system user, the system provider, and outside personnel. This development team is the entity in charge of setting up or designing the software development project and tailoring and implementing the selected processes.

The development team first designs the project. This action is affected by the product specification and constraints illustrated in Stage 1, but this impact falls outside the scope of the current research-in-progress; thus, it is not represented in Figure 2[1]. The developer team also tailors the selected software development processes (as determined by process selection and alignment in Stage 1). The property of "Selected Process" is "WPD" (Work Product Description), which is the target of the "Tailor and Implement" action.

When the development team tailors and implements the selected software development processes, it builds the software system. As a result, the selected processes affect project performance. This stage captures the other two characteristics identified by Germonprez et al. (2007) as definitional for a tailorable system: recognizable environment and component architecture (p. 355). The tailoring action is affected by the characteristics of the development team and the project, as shown by the solid arrows in Figure 2.

Cameron (2002) illustrated the practices in IBM to configure the software development processes, and explained the concept of WPD. Cameron also proposed several factors affecting the tailoring process. Based on Cameron's work, we propose the following hypotheses.

[1] To capture this aspect, we could combine Figures 1 and 2 and add arrows from "product" properties to the "design" action

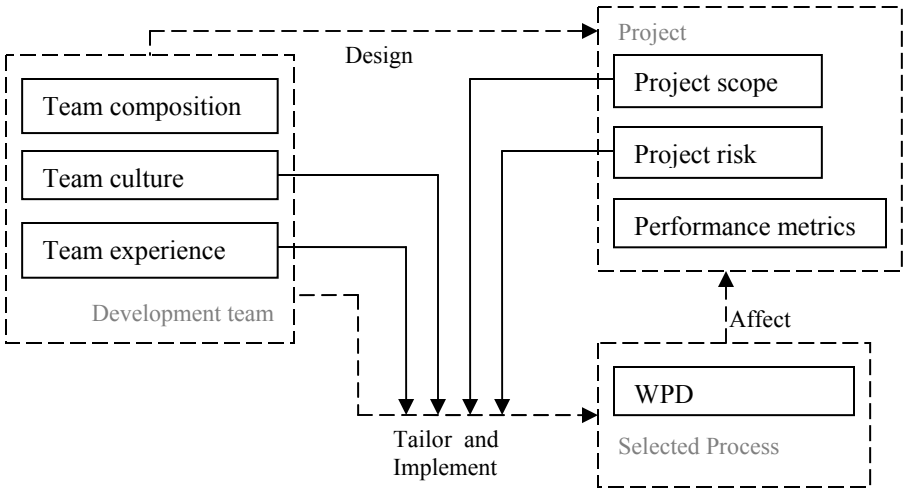

Figure 2: Project design and selected process tailoring and implementation

- If the team has more experience, the level of tailoring will be higher;
- If the project scope is specific rather than general, the level of tailoring will be lower;
- If the project risk is high, the level of tailoring will be lower.

Also, Iivari and Huisman (2007) conducted an empirical study to illustrate the relationship between organizational culture and the deployment of a systems development methodology. Applying their model to our context, we propose the following hypothesis:

- If the team has a rational culture rather than a hierarchicavl culture, the leel of tailoring will be higher.

3.3 Stage 3: Monitor Project and Provide Feedback

In this stage, the system user, system provider, and development team monitor the project performance and provide feedback based on the performance metrics of the project, as shown by the solid arrows in Figure 3. Based on the feedback, the system user and provider may modify the product design and process selection accordingly (as illustrated in Stage 1). The system user and system provider's action may also necessitate that the developer team modify the project design and process tailoring (as illustrated in Stage 2). Therefore, this stage expands Germonprez et al.'s (2007) theory of tailorable technology as an isolated single piece, and captures the continuous and cyclic nature of the software development procedure.

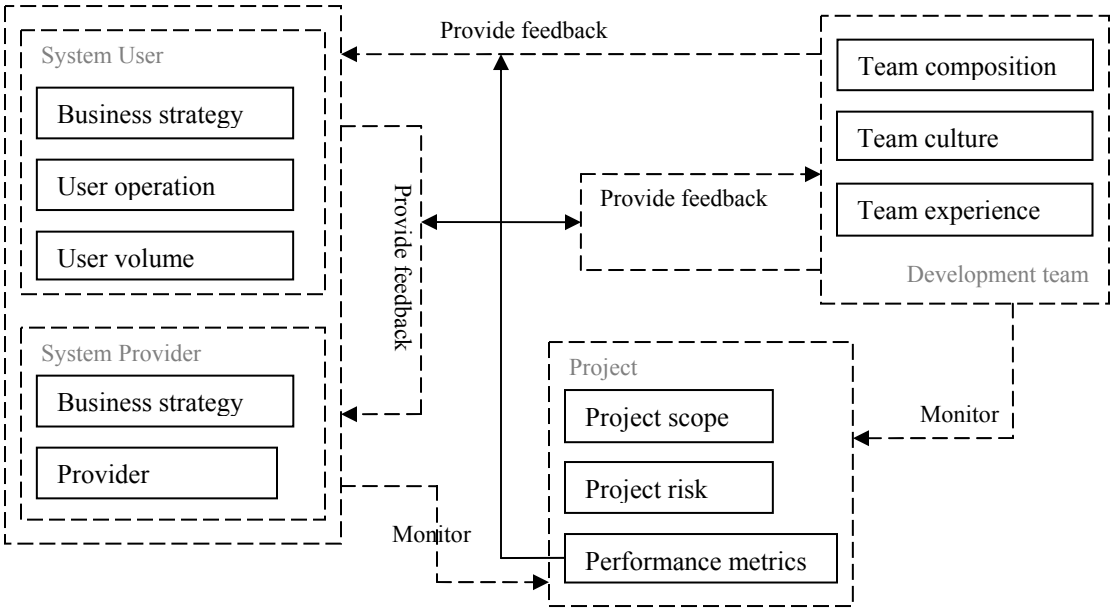

Figure 3: Monitor project and provide feedback

4. PLAN FOR NEXT STEPS

The model presented above captures the essence of our research focus and provides a basic foundation for more fully understanding the tailoring of software development.

The first step we plan to take next is to further flesh out our theoretic model. We also plan to more fully articulate how our work makes a significant contribution to the current literature on tailoring software development. More specifically, we plan to more fully review areas of limited understanding regarding the tailoring of software development and indicate more completely how our model addresses those limitations.

Once the theoretic model is constructed, we plan to conduct an empirical study to test our hypotheses and offer practical insights based on the results.

5. REFERENCES

[1] Adler, P. S. 2005. The evolving object of software development. Organization. 12, 3 (2005), 401-435.

[2] Barney, J. 1991. Firm resources and sustainable competitive advantage. Journal of Management. 17, 1 (1991), 99-120.

[3] Caldeira, M. and Ward, M. 2003. Using resource-based theory to interpret the successful adoption and use of information systems and technology in manufacturing small and medium-sized enterprises. European Journal of Information Systems. 12, 2 (2003), 125-139.

[4] Cameron, J. 2002. Configurable development processes. Communications of the ACM. 45, 3 (2002), 72-77.

[5] Forrester, E., Ed. 2006 A Process Research Framework, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

[6] Germonprez, M., Hovorka, D. S. and Collopy, F. 2007. A theory of tailorable technology design. Journal of the Association for Information Systems. 8, 6 (June 2007), 351-367.

[7] Gregor, S. and Jones, D. 2007. The anatomy of a design theory. Journal of the Association for Information Systems. 8, 5 (May 2007), 312-335.

[8] Iivari, J., Hirschheim, R., and Klein, H. K. 2000-2001. A dynamic framework for classifying information systems development methodologies and approaches. Journal of Management Information Systems. 17, 3 (Winter 2000-2001), 179-218.

[9] Iivari, J. and Huisman, M. 2007. The relationship between organizational culture and the deployment of systems development methodologies. MIS Quarterly. 31, 1 (2007) 35-58.

[10] Nidumolu, S. R. and Knotts, G. W. 1998. The effects of customizability and reusability on perceived process and competitive performance of software firms. MIS Quarterly. 22, 2 (June 1998), 105-137.

[11] Paulk, M., Curtis, B., Chrissis, M., and Weber, C. 1993. Capability maturity model, version 1.1. IEEE Software. 10, 4 (1993), 18-27.

[12] Slaughter, S. A., Levine, L., Ramesh, B., Pries-Heje, J., and Baskerville, R. 2006. Aligning software processes with strategy. MIS Quarterly, 30, 4 (December 2006), 891-918.

[13] Squire, B., Brown, S., Readman, J., and Bessant, J. 2006. The impact of mass customisation on manufacturing trade-offs. Production and Operations Management. 15, 1 (Spring 2006), 10-21.

[14] Thong J. 2001. Resource constraints and information systems implementation in Singaporean small businesses. Omega. 29, 2 (2001), 143-156.

[15] Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. 1992. Building an information system design theory for vigilant EIS. Information Systems Research. 3, 1 (1992), 36-59.

[16] Xu, P. and Ramesh, B. 2007. Software process tailoring: An empirical investigation. Journal of Management Information Systems. 24, 2 (Fall 2007), 293-328.