Investigating the Motivating Potential of Software Development Methods: Insights from a Work Design Perspective

Adarsh Kumar Kakar
Alabama State University
akakar@alasu.edu

Abstract

In this study we view Software Development Methods (SDMs) through the perspective of work design. The objective is to gain understanding into the inherent potential of the two major paradigms of software development, the Plan-driven and Agile methods, to motivate team members of software development projects. The work design perspective is relevant for investigating the motivating impacts of SDMs. The increasing popularity of people focused Agile methods over process focused Plan-driven methods in the 2000s mirror the increasing popularity of non-Taylorist work designs over Taylorist work designs in the 1980s. Work design concepts in parallel disciplines such as manufacturing evolved from the Taylorist focus on specialization and time and motion studies to adopting autonomous self-managed teams and creating employee programs like quality circles. Gleaning concepts from the widely accepted Job Characteristic Model (Hackman and Oldham, 1976) for work design, and the Self-Determination Theory (Deci and Ryan, 1985), one of the most prominent theories of motivation, provided useful insights. Together they could explain and predict the differential impacts of plan-driven and Agile methods of software development on team member motivation.

Keywords: Job Characteristic Model, Self-Determination Theory, Software Engineer Motivation, Software Engineer Job Satisfaction

Introduction

Motivating software engineers is an important goal of project managers. It is considered a means of improving productivity and reducing turnover (Franca, Sharp and Da Silva, 2014). Motivated engineers are more engaged (Franca, Sharp, and Da Silva, 2008) and motivation is widely acknowledged as a critical factor of success for software projects (Procaccino and Verner, 2006; Beecham, Baddoo, Hall, Robinson, and Sharp, 2008). Considering that one of the most contentious questions in software engineering is the choice of software development method (Johnson, Ekstedt and Jacobson, 2012), it is important to examine its impacts on motivation of software engineers before making a choice.

Currently there is a gap in Information Systems (IS) literature. The choice of the software development method (SDM) has been primarily driven by the customer and the product related factors (see Austin and Devin, 2009). While these factors are important, in this study we specifically examine how the choice of SDM might impact motivation of software development team members. A method is essentially a systematic way of performing a task or doing work. Therefore viewing software development methods (SDMs) from the perspective of work design might provide useful insights into their motivating potential by providing access to the vast existing literature on work design and making them available to the newer software engineering discipline. Since the industrial revolution, work design theories have been useful to describe and explain the behaviors of employees (e.g. Hackman and Oldham, 1974). Work design is known to affect employee psychosocial and work outcomes such as performance, turnover and absenteeism, job satisfaction, work motivation, stress, and burnout (e.g., Parker and Wall, 1998).

“Work design describes how jobs, tasks, and roles are structured, enacted, and modified, as well as the impact of these structures, enactments, and modifications on individual, group, and organizational outcomes” (Grant and Parker, 2009). Parallel disciplines such as Manufacturing and New Product Development have gainfully adopted work design theories and practices in their quest of improved psychosocial and work outcomes. Further, work design theories have been the harbinger of evolution in manufacturing paradigms, from mass production to lean manufacturing to Agile manufacturing. A similar trend in evolution of software development methods can be seen with the introduction of the “software factory” principles and concepts for mass production of software components in the late 1960s (McIlroy, 1968; Bemer, 1969), lean software development in the 1990s (Freeman, 1992) and Agile principles in the 2000s (The Agile Manifesto, 2001).

Keeping this context in view, in this study, we examine the relevance of applying work design concepts and theories in understanding the motivating potential of software development methods. We found the Job Characteristics Model (JCM), one of the most elaborate and widely accepted theories of work design (Kiggundu, 1981), appropriate for this purpose. Together with the Self-Determination Theory (SDT), the JCM could not only explain and predict the differential impacts of SDMs on team member motivation but also provide insights into why certain project practices work. Further, it could provide guidance for tailoring project processes and practices suitably to enhance affective and work outcomes.

Literature Review

Overview of Evolution in Work Design

Work design research began with the economic perspectives on the division of labor and task specialization (Babbage, 1835; Smith, 1776). Adam Smith (1776)
suggested division of labor by breaking down complex jobs into simpler jobs as a way of enhancing performance. Expanding on these ideas Charles Babbage (1835) pointed out the added advantages of job simplification such as requirement of less skilled and hence cheaper labor. However, it was not until the early 20th century, that the design of work caught the attention of organizational scholars. Taylor (1911, 1947) analyzed tasks into their minutest details and arrived at a standardized process; the one best way to do the job (Kanigel, 1997), just as Eli Whitney analyzed a musket into its smallest parts and made a machine to manufacture each part (Mirskey and Nevins, 1952). Together the ideas of Babbage, Smith, Whitney, Taylor and Ford (of moving assembly line) ushered in the era of mass production.

Later, the human relations movement was launched by researchers due to the satisfaction and motivation costs of division of labor and task specialization (Grant and Parker, 2009). The human relations movement initially focused on studying whether improving social and environmental conditions would enhance employee satisfaction, motivation, productivity and comfort (Mayo, 1933, 1945; Roethlisberger and Dickson, 1939). While mass production resulted in an improvement in the standard of living of society, it had deleterious psychological consequences for the workers. Repetitive jobs were found to be boring, tiring, dissatisfying and potentially damaging to mental health (Fraser, 1947; Walker and Guest, 1952). These costs of division of labor and task specialization diverted the focus of researchers to human issues at work.

Further, increasing uncertainty at the work place implied that the tayloristic approach of managers and industrial engineers defining and assigning jobs to the employees would no longer work. Uncertainty indicates inability to anticipate when problems will arise and/or lack of knowledge about how best to deal with them (Jackson, 1989). When uncertainty is low, events are predictable and the means of dealing with them known. It is therefore possible to determine and enforce the ‘one best way’ of doing the job. In contrast, where there is high uncertainty, the occurrence of problems and the means of solving them are less predictable, which implies greater autonomy should be provided to teams for structuring work to quickly adjust to changing environment.

In the following decades the roots of contemporary work design research was planted by scholars (Grant and Parker, 2009). Herzberg and colleagues suggested that to increase employee motivation and satisfaction, jobs should be enlarged and enriched (Herzberg, 1966; Herzberg, Mausner and Snyderman, 1967); the Tavistock scholars examined the interdependencies of social and technical systems (Trist and Bamforth, 1951; Walker and Guest, 1952); and Turner and Lawrence (1965) highlighted the importance of task attributes in shaping job behaviors and perceptions.

Richard Hackman, Edward Lawler and Greg Oldham argued that Taylorist job design is sub-optimal (Hackman and Oldham, 1976; Hackman and Lawler, 1971; Lawler, 1973; Porter, Lawler and Hackman, 1975). Enriched jobs, by encouraging workers to learn and innovate at work, increase the motivating potential of work. Motivated workers perform tasks more accurately and are more likely to find productivity innovations that engineers overlook. Hackman and colleagues expanded and synthesized previous concepts about work design into the Job Characteristics Model (JCM), by focusing on five structural job characteristics - task variety, task autonomy, feedback, task significance and task identity - that could enhance internal work motivation, satisfaction and performance by cultivating experiences of meaningfulness, responsibility, and knowledge of results. “The JCM became the dominant model of work design” (Grant and Parker, 2009).
Relevance of Work Design Perspective to SDMs

A similar trend was witnessed in the relatively young discipline of software engineering. Scientific management principles led to the development of Taylorist approaches such as the waterfall model (Royce, 1970) and its variants. These methods promoted strong conformance to plan through upfront requirements gathering and systems design, programming standards, code inspections and productivity metrics. They encouraged division of labor leading to specialized roles of business analysts, system architects, programmers and testes (Melnik and Maurer, 2006). Taylorist methods such as the waterfall method were a dominant software development paradigm until the 2000s.

Although a substantial improvement over “code-and-fix”, an approach used by programmers in the 1950s to develop software iteratively, Taylorist methods have issues of addressing customers’ real business needs and keeping with the development schedules. Under conditions of rapidly evolving customer needs, the approach of full requirements definition, followed by a long gap before those requirements are delivered did not seem to work. With increasing problem complexity, changing scope and requirements, and evolving technologies, developers, over time, came to realize that software development projects using this approach would not work under increased uncertainty.

Agile software development began, in the early 2000s, as countermovement to the Taylorist software development processes like the Waterfall Model or the V-Model (Fowler and Highsmith, 2001). Formal processes are perceived as responsible for management induced inefficiencies in the software development process (Anderson, 2005). Therefore the proclamation of the Agile manifesto in 2001 (Fowler and Highsmith, 2001) was welcomed by many in the software community and as a result the last decade saw a rapid increase in the popularity of the Agile methods as compared with the Taylorist methods such as the waterfall model and its variances. Agile development proponents question the assumption that change and uncertainty can be controlled through a high degree of advanced planning and have discovered inadequacies in formal design that follow systematic procedures dictated by rigid processes (Nerur, Mahapatra and Mangalraj, 2005). Like the human relations movement of the 1940s, people rather than process and responding to change rather than conformance to plan became the center of focus.

Although this trend in the software domain of leaning towards Agile development methodologies lagged the trend in favor of non-Taylorist or “enriched” work design in other domains such as manufacturing, there is a strong equivalence between these trends. This justifies the relevance of using work design theories, such as the JCM to SDMs, which represent a transition from the tayloristic paradigm. Applying work design theories such as the JCM to SDMs will help in providing an appropriate theoretical canvas to understand the motivating potential of SDMs.

The Job Characteristics Model

Work design attempts to influence motivation through work. In one of the earliest work in the area, Herzberg’s motivation-hygiene theory (Herzberg, 1966), a distinction is made between factors that are motivators and those that are hygiene. Motivators are thought to be work related factors such as challenging work, achievement, responsibility and personal competence. Hygiene factors are external factors such as company policies, supervisory practices, pay and working conditions. They are not part of the work itself and have no power to motivate the employee but their absence can lead to employee dissatisfaction.
The Herzberg theory specifies that a job will enhance work motivation and satisfaction only to the degree that "motivators" are designed into the work itself. Changes that deal solely with "hygiene" factors should not lead to increases in employee motivation. Herzberg proposed that jobs can be enriched to include motivators to enhance work motivation. Job enrichment is a type of job redesign intended to reverse the effects of tasks that are repetitive requiring little autonomy. The underlying principle is to expand the scope of the job with a greater variety of tasks, vertical in nature, that require self-sufficiency. However, the Herzberg theory does not specify how the presence or absence of motivating factors can be measured for existing jobs. This increases the difficulty of testing the theory in on-going organizations. It also limits the degree to which the theory can be used to diagnose jobs prior to planned change, or to evaluate the effects of work redesign activities after changes have been carried out (Hackman and Oldham, 1976).

To overcome these limitations, Herzberg’s work on job enrichment was further refined in Hackman, Oldham, Janson and Purdy (1975) using what they called as the Job Characteristics Model (JCM). They suggested that a work re-design - where workers have a chance to learn and deploy multiple skills with a high degree of autonomy and supportive feedback; have a sense of association with the work they perform; and feel they are making a significant contribution by developing meaningful products and not just performing narrow repetitive tasks - would lead to salutary work outcomes. More than two decades of empirical research on the JCM lead to the following conclusions (Parker, Wall and Cordery, 2001). First, the predicted collective effects of the core job characteristics on affective outcomes (Figure 1) have been largely supported (e.g. Parker and Wall, 1998). Second the link between the critical psychological states and the job characteristics (see Figure 1) has not been confirmed (Johns, Xie and Fang, 1992; Parker and Wall, 1998).

Job characteristics theory suggests that an individual will be motivated to work when jobs are designed to satisfy three critical psychological states. These include:

1. the need for meaningful work
2. the need to be responsible for work outcomes
3. the need for performance feedback.

These psychological states are affected by the following five job characteristics:

1. Skill Variety. The degree to which a job requires a variety of different activities in carrying out the work, which involve the use of a number of different skills and talents of the person.
2. Task Identity. The degree to which the job requires completion of a "whole" and identifiable piece of work; that is, doing a job from beginning to end with a visible outcome.
3. Task Significance. The degree to which the job has a substantial impact on the lives or work of other people.
4. Autonomy. The degree to which the job provides substantial freedom, independence, and discretion to the individual in scheduling the work and in determining the procedures to be used in carrying it out.
5. Feedback. The degree to which carrying out the work activities required by the job results in the individual obtaining direct and clear information about the effectiveness of his or her performance.

The critical psychological states in turn affect the motivation, job satisfaction and productive behaviors of employees resulting in increase in superior personal and work outcomes (see Figure 1).
Figure 1 - Job Characteristics Model

The Self-Determination Theory

Pinder (1998) defined work motivation as “a set of energetic forces that originates both within as well as beyond an individual’s being, to initiate work-related behaviour, and to determine its form, direction, intensity and duration” (p. 11). The SDT focuses explaining the “why” of behavior through insights into its nature. The underlying assumption is that “human beings are active, growth-oriented organisms who are naturally inclined toward integration of their psychic elements into a unified sense of self and integration of themselves into larger social structures” (Deci and Ryan, 2000, p. 229). The natural psychological growth and integration tendencies are susceptible to environmental and social conditions which can either disrupt or support propensities for self-determination (Ryan and Deci, 2002).

Consequently, the SDT distinguishes between lack of motivation (amotivation) and motivation. While motivation refers to intention to act or perform a task, amotivation represents having an intention to act passively (Deci and Ryan, 2002). Within motivation, the SDT distinguishes between intrinsic motivation and extrinsic motivation. Intrinsic motivation leads to behaviors that are performed for its own sake, in order to experience pleasure and satisfaction inherent in the activity (Deci and Ryan, 1987; Vallerand, 1997), regardless of the external benefits. Extrinsic motivation pertains to a wide variety of behaviors performed for reasons beyond those inherent in the activity itself. With focus on external rewards, such as improved job performance, pay, or promotions’ (Davis et al., 1992), the activity becomes a means to an end rather than an end in itself (Deci and Ryan, 1987).

These three primary types of motivation represent a continuum (Ryan and Deci, 2002). Extrinsic motivation can be further classified into four types on a continuum.
with motivation controlled by external motivation at one end and self-regulated through acquired values and goals at the other (see Figure 2). Thus at one end of the continuum (see Figure 2) is amotivation in which individuals either act passively or lack the intention to perform an activity or task. Next is external regulation (extrinsic motivation type 1) which involves performing a task only to obtain a reward. Then we have introjected regulation (extrinsic motivation type 2), that involves performance of task through self-worth contingencies such as guilt or self-esteem. Next on the continuum is identified regulation (extrinsic motivation type 3), which involves performance of a task because of identification with its value or meaning. Then there is integrated regulation (extrinsic motivation type 4), which involves identifying with the value of a task or activity to the extent that it becomes an integral part of the individual’s sense of self. This is the most internalized form of extrinsic motivation and therefore regarded as autonomous. Finally, we have intrinsic motivation at the other end of the continuum which is considered inherently autonomous motivation. Whereas identified, integrated and intrinsic motivations are considered self-determined motivations, amotivation, external and introjected motivations are considered as non-self-determined motivations.

Figure 2 - Motivation Continuum
Note: Gagné and Deci, 2005.

Hypotheses Development

The SDT (Ryan and Deci, 2000) identifies three fundamental human needs, the need for autonomy (De Charms, 1968; Deci, 1975), competence (Harter, 1978; White, 1959) and relatedness (Baumeister and Leary, 1995; Reis, 1994), that have a positive effect on intrinsic motivation. Cognitive Evaluation Theory (CET) (Deci and Ryan, 1985) a sub-theory within the SDT suggests that people need to feel competent and autonomous, and therefore factors that promote feelings of competence and autonomy enhance intrinsic motivation. Further, factors that diminish these feelings undermine intrinsic motivation and result in people either controlled by contingencies or amotivated.

This is consistent with the JCM (e.g., Gagné and Deci, 2005; Gagné, Sénécal and Koestner, 1997) which proposes that
increasing internal work motivation involves designing jobs that (1) provide variety, have positive impact on the lives of others and require completion of a whole rather than narrow tasks (relatedness), that (2) provide considerable discretion and freedom to the employee in accomplishing tasks (autonomy) and that (3) provide meaningful performance feedback (promote competence). These job characteristics correspond to promoting feelings of relatedness, competence and autonomy leading to self-determined motivation (e.g., Gagne’, Sénécal and Koestner, 1997). Internal work motivation of JCM is a concept similar to self-determined or autonomous motivation of SDT which includes a mix of intrinsic motivation and internalized extrinsic motivation such as integrated and identified extrinsic motivators (Gagne and Deci, 2005).

Autonomy is the degree to which the job provides freedom, independence, and discretion to the employee in scheduling the work and in determining the procedures to be used in carrying it out (Hackman and Oldham, 1976). Hackman and Oldham (1976) suggested that when employees have freedom to schedule their work and decide on procedures it increases the motivating potential of work. Studies have shown that autonomy is rated higher than any other job characteristic by knowledge workers (Cheney, 1984; Goldstein and Rockart, 1984; Janz et al., 1997). The motivational effects of autonomy are largely supported by empirical findings (Parker and Wall, 1998; Roberts, Hann and Slaughter, 2006; Beecham, Baddoo, Hall, Robinson, and Sharp, 2008).

Job enrichment through higher levels of job characteristics makes the work non-repetitive and interesting (McKnight, Philips and Hardgrave, 2009). If an employee performs a task because he enjoys doing so then the task is an intrinsic motivator. When people engage an activity because it is interesting and fun, they are voluntarily performing the activity. In contrast, being controlled to perform an activity involves acting with a sense of pressure, a sense of having to engage in the actions unwillingly.

The variety of skills required to perform the job makes it challenging. Studies showed that optimally challenging activities were highly intrinsically motivating (e.g., Danner and Lonky, 1981). By contrast, repetitive narrow jobs were found to be boring, tiring, dissatisfying and potentially damaging to mental health (Fraser, 1947; Walker and Guest, 1952). Task identity enhances the meaningfulness of work and task significance its importance. Therefore enriched jobs are designed in such a way that workers are involved in completing “whole” tasks that require a variety of skills and which are seen by them as having a significant impact on the lives of others (e.g. the customer or the user). Feedback (Deci, 1971) facilitates motivation by promoting a sense of competence as people feel responsible for their successful performance through knowledge of results (Fisher, 1978; Ryan, 1982).

Thus the 5 job characteristics of the JCM will be positively related to team members’ self-determined (identified, integrated and intrinsic) motivation by fulfilling their fundamental needs for autonomy, competence and relatedness. Testing this first set of 5 hypotheses will help assess the relevance of the work design in motivating software developers

Hypothesis 1: The degree of skill variety required to perform tasks in software development projects is positively related to self-determined motivation (identified, integrated and intrinsic)

Hypothesis 2: The degree of team members’ task identity in software development projects is positively related to self-determined motivation (identified, integrated and intrinsic)

Hypothesis 3: The degree of team members’ perception of significance of tasks in software development projects is positively related self-determined motivation (identified, integrated and intrinsic)
Hypothesis 4: The degree of team members’ autonomy in organizing work and performing tasks in software development projects is positively related to self-determined motivation (identified, integrated and intrinsic).

Hypothesis 5: The degree of feedback provided to team members in Software development projects is positively related to self-determined motivation (identified, integrated and intrinsic).

Plan-driven software development methods typically deploy specialized skill-based roles, with individual team members requiring less skill variety to accomplish jobs. Detailed planning is done of entire software development lifecycle activities including requirements gathering, design, construction, testing and project coordination and management activities and specialized people handle each of these tasks. The allocation of work specifies “not only what is to be done but how it is to be done and the time allowed for doing it” (Chau, Maurer and Melnik, 2003). This reduces the autonomy of employees and shifts the focus from individuals and their creative abilities to the processes themselves.

On the other hand Agile methods emphasize and value individuals and interactions over processes. Agile methods are people-centric, recognizing the value competent people and their relationships bring to software development (Nerur, Mahapatra and Mangalaraj, 2005). People issues are at the heart of the Agile movement (Boehm and Turner, 2005). The Agile team works by placing people physically closer, replacing documents with talking in person and at whiteboards, improving the team’s amicability and its sense of community (Cockburn and Highsmith, 2001). Agile methods recognize that periodic formalized feedback on software development can be ambiguous making frequent interpersonal feedback necessary to interpret performance.

Also, a user representative is often a part of the Agile software development project. Unlike Plan-driven approaches in which the customer is mainly involved during the early phase of the project and during product acceptance, Agile methods involve the customer throughout the whole development process (Paetsch, Eberlein and Maurer, 2003). Grant (2007, 2008) proposed that when employees work in jobs that interact with the beneficiaries (such as users) then it enhances their task significance. Further, in traditional work environments it is assumed that managers are responsible for structuring jobs to be executed by employees (Hackman and Oldham, 1976, 1980). However as uncertainty increases, such as in software development, due to rapid technological and market changes coupled with time-to-market pressures, it becomes increasingly difficult for managers to plan tasks (Griffin et al., 2007). Under such circumstances roles and tasks need to change rapidly to address emergent demands and new opportunities (Ilgen and Hollenbeck, 1991). As a result team members have greater autonomy and self-organizing teams have replaced static work structures of Plan-driven methods.

Agile methods disavow the deterministic/mechanistic view of problem solving to a dynamic process characterized by iterative cycles and the active involvement of all stakeholders. Big upfront design plans and extensive documentation are of little value to practitioners of Agile methods (Nerur, Mahapatra and Mangalaraj, 2005). Unlike the Taylorist methods, where the cycle time between requirements gathering and product release is typically very long, the gaps between customer requirements and implementation into the product in Agile projects are narrowed in rapid cycles. Practices such as paired programming, retrospectives, and stand-up meetings...
provide rapid feedback to developers compared to Plan-driven methods. The focus on developing working products rather than paper artifacts and components of plan-driven methods enhances task identity. These differences in job characteristics for Agile and Plan-driven methods and their empirical support in literature are summarized in Table 1. This led to a second set of 5 hypotheses that the Agile methods of software development will be higher on the 5 job characteristics compared with Plan-driven methods.

<table>
<thead>
<tr>
<th>Table 1 - Summary of differences in SDMs from the JCM perspective</th>
<th>Plan-driven methods of software development</th>
<th>Agile methods of software development (Beck, 1999; Scrum Alliance, 2008)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Skill Variety</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Specialized Skills based Roles: Designers, Programmers, Testers (Melnik and Maurer, 2006)</td>
<td>Multi-skilled employees, Role Flexibility (Nerur, Mahapatra and Mangalraj, 2005)</td>
<td></td>
</tr>
<tr>
<td><strong>Task Identity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Lower Task Identity. Tasks include project artifacts such as project plan, requirements specification, design documents, code and test reports; final product appears at the end of a long cycle of sequential phases (Cusumano, 1989; Paulk, Weber, Curtis and Chrissis, 1995; Hoda, Noble, Marshall, 2011)</td>
<td>Focus on “whole task” such as developing working products in short iterative cycles. “Whole tasks” contribute to greater task identity</td>
<td></td>
</tr>
<tr>
<td><strong>Task Significance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Building product or product components according to specifications although important have comparatively lower task significance</td>
<td>Focus on developing working products that provide competitive advantage to customers (beneficiaries) enhances task significance (Dyba and Dingsoyr, 2008; Moe, Dingsoyr, and Dybå, 2009)</td>
<td></td>
</tr>
<tr>
<td><strong>Task Autonomy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Fewer feedback mechanisms. Hand-offs from one phase to the next, periodic reviews and inspections, project closure meetings provide sporadic feedback (Royce, 1970; Moe, Aurum and Dybå, 2012)</td>
<td>Practices such as Open Work Space, Co-location of development team, Daily Stand up Meetings, Paired Programming, User Representative on the development team, sprint reviews, project retrospectives provide continuous and intense feedback (Beck, 1999; Scrum Alliance, 2008)</td>
<td></td>
</tr>
</tbody>
</table>

Hypothesis 6: The team members of Software development projects using Agile methods perform tasks requiring significantly greater degree of skill variety than those using Plan driven methods.

Hypothesis 7: The team members’ perception of task identity is significantly greater for software development projects using Agile methods than those using Plan driven methods.

Hypothesis 8: The team members’ perception of significance of tasks is significantly greater for software development projects using Agile methods than those using Plan driven methods.
Hypothesis 9: Software development projects using Agile methods provide significantly greater task Autonomy than those using Plan driven methods.

Hypothesis 10: Software development projects using Agile methods provide significantly greater feedback to its team members than those using Plan driven methods.

With tasks becoming more socially embedded than at any other time in the past, work design researchers recognize that work is inextricably intertwined with interactions among team members and interpersonal relationships (Grant and Parker, 2009). According to a typology of interdependence (see Figure 3) by Tesluk et al. (1997) the degree of interdependence increases from pooled to sequential to reciprocal to intensive. Pooled interdependence does not involve any interaction between team members. Performance of the group is an aggregation of individual team member’s performance. In sequential interdependence, work flows unidirectionally from one member to another. Reciprocal interdependence is similar to sequential except that the workflow is bidirectional. In intensive interdependence the entire group must interact with each other to accomplish group goals.

![Figure 3 - Typology of Interdependence](image)

Note: adopted from Tesluk et al., 1997.
Plan driven methods of software development such as the waterfall method and its variants promote conformance to plan and encourage division of labor leading to specialized roles of business analysts, system architects, programmers and testers (Melnik and Maurer, 2006). Tasks are process-driven, team members have lesser autonomy and points of employee interfaces are few. Sequential interdependence predominates as can be seen from Figure 4 for waterfall model (Royce, 1970). Typically, testers interact with coders but not with designers, designers interact with requirement gatherers but not with system implementers.

By contrast the agile methods deploy self-managing teams. Teams and its members have more autonomy. Work in Agile teams are not defined by industrial engineers and assigned by supervisors but by autonomous self-organizing teams through mutual adjustment amongst team members (Cockburn and Highsmith, 2001; The Agile Manifesto, 2001). Their activities are guided by a common vision and controlled by a democratic system of control called concertive control. Concertive control was argued to be a viable alternative to the bureaucratic control marked by a hierarchical system with rational-legal rules rewarding compliance and punishing deviations (Baker, 1993). Practices such as pair programming, planning game, daily stand-up meeting and frequent integration of code to produce working products in quick iterations represent intensive interdependence among group members in accomplishing tasks as can be seen from Figure 5 for Extreme Programming (Beck, 1999).
Concertive control of self-organizing teams is more likely to result in autonomous (self-determined motivation) of team members while the hierarchical command control structures are more likely to lead to controlling (non-self-determined) motivation of team members. Further, interpersonal feedback is critical as software development is a collaborative activity requiring continuous and intense interaction. Periodic formalized reports into software development activities can be ambiguous making interpersonal feedback from multiple perspectives necessary to interpret performance. Additionally, frequent feedback is necessary to provide information about the extent to which the individual and team efforts meet the objectives of the software development project and expectations of customers. Agile practices such as sprint reviews and daily stand-up meetings are very useful in this regard. The above practices satisfy the fundamental human needs for relatedness, autonomy and competence, thereby positively impacting the self-determined motivations of team members (Deci and Ryan, 1985).

This discussion leads us to the following hypotheses:

Hypothesis 11: Software Development projects using Agile Methods will provide significantly higher self-determined motivation (identified, integrated and intrinsic) than those using Plan-driven method

Hypothesis 12: Software Development projects using Agile Methods will provide significantly lower non-self-determined motivation (amotivation, external and introjected) than those using Plan-driven method

The self-determination continuum (Figure 2) is useful in predicting outcomes such as job performance, employee engagement, job satisfaction, retention and subjective wellbeing (Gagne’ and Forest, 2008) necessary for optimal functioning (Tremblay, Blanchard, Taylor, Pelletier and Villeneuve, 2009). The SDT postulates that among self-determined motivations, intrinsic motivation leads to the most positive consequences, followed by integrated and identified regulations. Among non-self-determined motivations, Introjected and external
regulations lead to negative outcomes. Amotivation results in the most negative consequences (Vallerand and Ratelle, 2002). These negative outcomes include counterproductive performance and employee withdrawal (Pelletier, Vallerand, Blais, Brière and Green-Demers, 1996).

Multiple studies provided empirical support for the self-determined continuum. Blais et al. (1993) investigated the effect of the self-determination continuum in a work setting and provided empirical support for it. Their results indicated that while external and introjected regulations were associated with physical and mental health problems and emotional exhaustion, (also see Houkes, Jassen, de Jonge, and Bakker, 2003), self-determined motivation displayed positive associations with job and life satisfaction (also see Locke and Latham, 2004). Therefore considering that team members of projects using agile methods will have a significantly higher level of self-determined motivation than team members of projects using plan-driven methods (Hypothesis 11) in conjunction with the above arguments leads us to the following hypotheses in line with the predictions of the JCM (see Figure 1):

Hypothesis 13: Team members of Software development projects using Agile methods will have a significantly higher satisfaction than team members of software development projects using Plan-driven methods.

Hypothesis 14: Team members of Software development projects using Agile methods will show a significantly higher level of performance than team members of software development projects using Plan-driven methods.

Hypothesis 15: Team members of Software development projects using Agile methods will show a significantly lower level of work exhaustion than team members of software development projects using Plan-driven methods.

Method

Study Setting and Design

To test the relevance of the JCM to SDMs we conducted a paper and pencil based survey with development team members of 56 software projects. The developers were employees of a large Indian software development organization with operations across the globe. The 56 projects included software development for 32 companies across 8 countries in North America, Europe and Asia. The type of projects included 23 which were characterized by Project Managers as Waterfall method, 4 V-method, 17 Extreme programming, 7 Scrum, 1 Crystal methodologies, 2 Dynamic Software development method (DSDM) and 2 Feature Driven Development (FDD). The study was completed over a 5 year period involving 321 developers who answered a survey questionnaire at the end of completion of their projects and represented the response from 86 % of developers who participated in the 56 development projects.

Variables used in the Study

The independent variables were skill variety, autonomy, feedback, task significance and task identity. The dependent variables were self-determined, non-self-determined motivation, satisfaction, work exhaustion and performance. Although the aim of the study was to fully assess the relevance of the JCM (see Figure 1) to software development not all dependent variables could be tested. For example, the impact of these job characteristics on job turnover could not be examined because ascribing the turnover to employee experience with a project was difficult to ascertain. Also the impact of job characteristics on absenteeism could not be studied as not all team members worked full-time on a project and data to the level of specificity required for this analysis was not maintained by the projects. In lieu of these variables, the impact of job characteristics on work exhaustion was examined, as work exhaustion is associated with absenteeism.
and job turnover (Moore, 1991). Tested measures from prior literature were adapted to capture data pertaining to these variables:

The Job Diagnostic Survey (JDS) (Hackman and Oldham, 1974) consisting of list of items for all 5 job characteristics was used. A sample item from this scale is “To what extent do managers or coworkers let you know how well you are doing on your job”. For a complete list of items see Appendix A.

**Self-determined motivation.** Self-determined motivation is measured by summing the means of each of the three self-determined subscales (Tremblay, Blanchard, Taylor, Pelletier and Villeneuve, 2009): intrinsic motivation, integrated regulation and identified regulation. A sample item from this scale is getting response to the question why do you do your work: “For the satisfaction I experience from taking on interesting challenges.”

**Non-Self-Determined Motivation.** Non-Self-determined motivation is measured by summing the means of each of the three non self-determined subscales: amotivation, external and introjected motivations (Tremblay, Blanchard, Taylor, Pelletier and Villeneuve, 2009). A sample item from this scale is getting response to the question why do you do your work: “Because it will allow me to earn money.”

**Satisfaction.** The general satisfaction scale (Hackman and Oldham, 1974) was used for measuring general satisfaction. A sample item from this scale is: “I am satisfied with the kind of work I did on the project”.

**Work Exhaustion.** The McKnight, Philips and Hardgrave (2009) scale was used for measuring the work exhaustion. A sample item from this scale is: “I feel burned out from my work”.

**Performance.** The performance of the subjects was measured on 3 criteria by his immediate supervisor. A sample item from this scale is “The team member completed all work assigned to him on time”.

For a complete list of items used in the measures of Self-Determined Motivation, Non-Self-Determined Motivation, Satisfaction, Work Exhaustion and Performance scales please see Appendix B. These measures used a 9-point Likert scale with anchors of 1 (strongly disagree) and 9 (strongly agree). A review of the literature indicates that expanding the number of choice-points beyond 5- or 7-points does not systematically damage scale reliability, yet such an increase does increase scale sensitivity (Cummins and Gullone, 2000). Scale items were averaged to create an overall value for each construct. Responses were coded such that high levels of the constructs are represented by high values. Some items were reverse coded.

**Subjects**

The subjects were 21-43 year olds, 186 males and 135 females. Their average age was 28.4 years with an average software development experience of 6.5 years.

**Procedure**

Subjects answered a paper-and-pencil based survey that captured data on independent variables, Job Characteristics, and, dependent variables, Self-determined Motivation, Non-Self-determined Motivation, Work Exhaustion, Satisfaction and Performance with the software development project. The questionnaire items listed were scrambled. Data on independent variables, Job Characteristics was collected from the subjects in the first round of the study. Data on the dependent variables, Self-Determined Motivation, Non-Self-Determined Motivation, Work Exhaustion and Satisfaction was collected in the second round a week later. Previous research demonstrates that the temporal separation between measures reduces potential effects due to Common Method Variance (Sharma, Yetton and Crawford, 2009). Data on subject Performance on the project was captured from the immediate supervisor of the subject.
Method of Analyses

To establish reliability and validity of the measures used in the study factor analysis was performed and internal reliabilities and correlation matrix of the measures were examined. To assess the difference in the independent variable, job characteristics, and the dependent variables Self-determined Motivation, Non-Self-determined Motivation, Work Exhaustion, Satisfaction and Work Performance between team members working on Agile software development projects and Plan-driven software development projects t-tests were performed. Multiple Regression Analyses was done using the 5 job characteristic as independent variable and Self-determined Motivation, Non-Self-determined Motivation, Work Exhaustion, Satisfaction and Work Performance as dependent variables to estimate the significance of the impacts of each of the job characteristic on the dependent variables.

Results and analyses

The factor analysis procedure was done using IBM© SPSS© Statistics Version 19. Dimension reduction was performed on the data pertaining to the 6 measurement scales. The results of Varimax rotation showed that the 6 factors extracted represented each of the six scales. All items of a scale (Motivation: 1 to 18, Job Characteristics: J1 to J5, Satisfaction: S1 to S3, Work Exhaustion: W1 to W4 and, Performance: P1 to P3) loaded on the respective factors (highlighted in bold in Appendix C). Convergent and discriminant validity between scales were evident (see Appendix C) by the high loadings within factors (> .50) and no cross loadings (> .40) between factors. The internal reliabilities of the scales used in the study: job characteristic, self-determined motivation, non-self-determined motivation, satisfaction and performance were then examined. As can be seen from the Table 2, the alpha reliabilities are all greater than .70.

<table>
<thead>
<tr>
<th>Table 2 - Internal Reliability of Scales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the scale</td>
</tr>
<tr>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Job Characteristics</td>
</tr>
<tr>
<td>Self-Determined Motivation</td>
</tr>
<tr>
<td>Non-Self-Determined Motivation</td>
</tr>
<tr>
<td>Satisfaction</td>
</tr>
<tr>
<td>Work Exhaustion</td>
</tr>
<tr>
<td>Performance</td>
</tr>
</tbody>
</table>

Table 3 provides the means and standard deviations of the data collected in this survey. From the correlation between variables in Table 4 it is clear that none of the correlations are too high (> 0.65) demonstrating that each scale is adding something new.

<table>
<thead>
<tr>
<th>Table 3 - Descriptive Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Job Characteristics</td>
</tr>
<tr>
<td>Self-Determined Motivation</td>
</tr>
<tr>
<td>Non-Self-Determined Motivation</td>
</tr>
<tr>
<td>Satisfaction</td>
</tr>
<tr>
<td>Work Exhaustion</td>
</tr>
<tr>
<td>Performance</td>
</tr>
</tbody>
</table>
### Table 4 - Correlations between measures

<table>
<thead>
<tr>
<th>Measures</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Characteristics</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-Determined Motivation</td>
<td>0.457**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Self-Determined Motivation</td>
<td>0.258*</td>
<td>0.107</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Satisfaction</td>
<td>0.392**</td>
<td>0.219*</td>
<td>0.238*</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work Exhaustion</td>
<td>-0.309*</td>
<td>-0.261*</td>
<td>-0.118</td>
<td>-0.231*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>0.236*</td>
<td>0.221*</td>
<td>0.162</td>
<td>0.194*</td>
<td>-0.095</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes: * p < .05  ** p < .01  *** p < .001

We next examine the results of multiple regression analyses. The 5 job characteristics were regressed on outcome variables. The results in Table 5 show that overall the job characteristics significantly and positively impacted self-determined motivation, non-self-determined motivation, satisfaction and performance of team members and negatively impacted work exhaustion. However, task identity did not significantly impact non-self-determined motivation and performance of team members and skill variety and feedback did not significantly impact work exhaustion and performance respectively. Thus Hypotheses 3 and 4 were fully supported and Hypotheses 1, 2 and 5 were partially supported.

### Table 5 - Regression Results of job Characteristics

<table>
<thead>
<tr>
<th>Job Characteristics</th>
<th>Self-Determined Motivation</th>
<th>Non-Self Determined Motivation</th>
<th>Satisfaction</th>
<th>Work Exhaustion</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill Variety</td>
<td>.834**</td>
<td>0.687**</td>
<td>0.441**</td>
<td>-0.501**</td>
<td>0.123</td>
</tr>
<tr>
<td>Task Identity</td>
<td>.782**</td>
<td>0.137</td>
<td>0.150</td>
<td>-0.623**</td>
<td>0.219</td>
</tr>
<tr>
<td>Task Significance</td>
<td>.821**</td>
<td>0.333*</td>
<td>0.599**</td>
<td>-0.423**</td>
<td>0.475**</td>
</tr>
<tr>
<td>Autonomy</td>
<td>.925***</td>
<td>0.475**</td>
<td>0.656**</td>
<td>-0.476**</td>
<td>0.524**</td>
</tr>
<tr>
<td>Feedback</td>
<td>.791**</td>
<td>0.312**</td>
<td>0.731***</td>
<td>-0.141</td>
<td>0.324**</td>
</tr>
</tbody>
</table>

Notes: * p < .05  ** p < .01  *** p < .001

### Table 6 - Differences between the two methods on job characteristics

<table>
<thead>
<tr>
<th>Job Characteristics</th>
<th>Agile methods</th>
<th>Plan-Driven Methods</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill Variety</td>
<td>6.31</td>
<td>4.64</td>
<td>1.67***</td>
</tr>
<tr>
<td>Task Identity</td>
<td>5.89</td>
<td>5.04</td>
<td>0.85*</td>
</tr>
<tr>
<td>Task Significance</td>
<td>6.42</td>
<td>4.81</td>
<td>1.61**</td>
</tr>
<tr>
<td>Autonomy</td>
<td>7.37</td>
<td>5.96</td>
<td>1.41***</td>
</tr>
<tr>
<td>Feedback</td>
<td>6.65</td>
<td>5.18</td>
<td>1.47***</td>
</tr>
</tbody>
</table>

Notes: * p < .05  ** p < .01  *** p < .001
From Table 6 we can see that from the “job enrichment” perspective the Agile work design in general is superior to Plan-driven methods. The t-tests show that all the 5 job characteristics were significantly (p=0.05) higher for Agile methods than for Plan-driven methods. Thus Hypotheses 6-10 are found valid.

From Table 7 we can infer that except for non-self-determined motivation, the team members of projects adopting Agile methods performed significantly better than Plan-driven methods on self-determined motivation, performance, work exhaustion and satisfaction than team members of Plan-driven methods. No significant difference between the two methods was observed for team members’ non-self-determined motivation. Thus Hypotheses 11, 13, 14 and 15 were supported, but Hypothesis 12 was not supported.

<table>
<thead>
<tr>
<th>Table 7 - Employee and Project Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Project Outcomes</td>
</tr>
<tr>
<td>Agile Methods</td>
</tr>
<tr>
<td>Plan Driven Methods</td>
</tr>
<tr>
<td>Difference</td>
</tr>
<tr>
<td>Self-Determined Motivation</td>
</tr>
<tr>
<td>7.58</td>
</tr>
<tr>
<td>6.14</td>
</tr>
<tr>
<td>1.43**</td>
</tr>
<tr>
<td>Non-Self Determined Motivation</td>
</tr>
<tr>
<td>6.67</td>
</tr>
<tr>
<td>6.44</td>
</tr>
<tr>
<td>0.23</td>
</tr>
<tr>
<td>Satisfaction</td>
</tr>
<tr>
<td>7.63</td>
</tr>
<tr>
<td>6.19</td>
</tr>
<tr>
<td>1.44**</td>
</tr>
<tr>
<td>Work Exhaustion</td>
</tr>
<tr>
<td>3.41</td>
</tr>
<tr>
<td>5.86</td>
</tr>
<tr>
<td>-2.45***</td>
</tr>
<tr>
<td>Performance</td>
</tr>
<tr>
<td>6.28</td>
</tr>
<tr>
<td>5.17</td>
</tr>
<tr>
<td>1.11*</td>
</tr>
</tbody>
</table>

Notes: * p < .05  ** p < .01  *** p < .001

Discussion

The findings of the study show that overall the 5 job characteristics proposed in the model, skill variety, task identity, task significance, autonomy and feedback impact affective and work outcomes of software development projects. However, only task significance and autonomy were found to significantly impact all affective and work outcomes. The impact of task identity on satisfaction was not found to be significant (Table 8). Also, consistent with observations in the large scale by Hackman and Oldham (1976) survey with 7 organizations, a weak link between the 5 characteristics was observed for team member performance. Skill Variety and Task Identity were not found to impact performance significantly. The reason for this weak relationship may be that affective outcomes are causally more proximal to job characteristics than are behavioral outcomes such as performance (Hackman and Oldham, 1976).

In general, the job characteristics were able to clearly distinguish between Agile and Plan-driven methods of software development. The Agile methods scored higher on all the 5 job characteristics compared to Plan-driven methods (Table 6). Thus it was not a surprise to find significant difference between the impacts of Agile methods and Plan-driven methods on outcome variables (Table 7) in line with predictions of the JCM (Figure 1). Team members of Agile projects performed better than Plan-driven methods on outcome variables such as self-determined motivation, satisfaction, work exhaustion and performance thus overall supporting the relevance of applying the JCM to software development. However, the difference in non-self-determined motivation was not found to be significant.
Table 8 - Summary of Hypotheses (H1...H16) Supported (S) and Not Supported (NS in bold)

<table>
<thead>
<tr>
<th>Job Characteristics</th>
<th>Self-Determined Motivation</th>
<th>Non-Self-Determined Motivation</th>
<th>Satisfaction</th>
<th>Work Exhaustion</th>
<th>Performance</th>
<th>Agile - Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill Variety</td>
<td>S (H1)</td>
<td>S (H1)</td>
<td>S (H1)</td>
<td>S (H1)</td>
<td>NS (H1)</td>
<td>S (H6)</td>
</tr>
<tr>
<td>Task Identity</td>
<td>S (H2)</td>
<td>S (H2)</td>
<td>NS (H2)</td>
<td>S (H2)</td>
<td>NS (H2)</td>
<td>S (H7)</td>
</tr>
<tr>
<td>Task Significance</td>
<td>S (H3)</td>
<td>S (H3)</td>
<td>S (H3)</td>
<td>S (H3)</td>
<td>S (H3)</td>
<td>S (H8)</td>
</tr>
<tr>
<td>Autonomy</td>
<td>S (H4)</td>
<td>S (H4)</td>
<td>S (H4)</td>
<td>S (H4)</td>
<td>S (H4)</td>
<td>S (H9)</td>
</tr>
<tr>
<td>Feedback</td>
<td>S (H5)</td>
<td>S (H5)</td>
<td>NS (H5)</td>
<td>S (H5)</td>
<td>S (H5)</td>
<td>S (H10)</td>
</tr>
<tr>
<td>Agile- Plan</td>
<td>S (H11)</td>
<td>NS (H12)</td>
<td>S (H13)</td>
<td>S (H14)</td>
<td>S (H15)</td>
<td></td>
</tr>
</tbody>
</table>

To explore the reason for this finding, we performed a supplementary analysis of data on motivation (see Table 9). The results were rather surprising. Although, as expected, team members of Agile projects scored significantly lower on amotivation compared to team members of Plan-driven methods, they scored significantly higher on External and Introjected motivation. The reason could be the higher performance of Agile projects may have resulted in higher perceptions of self-worth of its team members resulting in higher introjected motivation and an increase in prospects of a rewarding career resulting in higher external motivation. Thus overall, Agile methods create conditions for promoting higher motivation among its team members whether it is intrinsic motivation (higher is better), extrinsic motivation (higher is better) or amotivation (lower is better).

Table 9 - Difference in Motivation between Plan-Driven and Agile Methods

<table>
<thead>
<tr>
<th>Motivation Types</th>
<th>Agile methods</th>
<th>Plan-Driven Methods</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic</td>
<td>7.97</td>
<td>5.34</td>
<td>2.63***</td>
</tr>
<tr>
<td>Integrated</td>
<td>7.27</td>
<td>6.34</td>
<td>0.93*</td>
</tr>
<tr>
<td>Identified</td>
<td>7.49</td>
<td>6.73</td>
<td>0.76*</td>
</tr>
<tr>
<td>Introjected</td>
<td>7.58</td>
<td>6.46</td>
<td>1.12**</td>
</tr>
<tr>
<td>External</td>
<td>7.44</td>
<td>6.18</td>
<td>1.26**</td>
</tr>
<tr>
<td>Amotivation</td>
<td>4.98</td>
<td>6.68</td>
<td>-1.7***</td>
</tr>
<tr>
<td>Extrinsic</td>
<td>7.45</td>
<td>6.43</td>
<td>1.02*</td>
</tr>
</tbody>
</table>

Notes: * p < .05  ** p < .01  *** p < .001

These findings have useful implications for practice. The JDS used in the study can be used for two general purposes (a) for diagnosing existing jobs to determine if (and how) they might be re-designed to improve employee motivation, productivity and
satisfaction; and (b) for evaluating the effect of job changes on employees—whether the changes derive from deliberate "job enrichment" projects or from naturally-occurring modifications of technology or work systems. The findings of the study suggest, in line with the JCM, that in order to improve psychological and behavioral outcomes of software development, all the five job characteristics should be developed. The level of each of the five job characteristics can be measured using this survey. If the psychological and work outcomes of a software development project do not meet defined goals, then interventions can be planned based on identification of those specific job characteristics that are most in need of improvement to achieve desired outcomes.

Further, the results obtained in the study suggest that the JCM can be also used to provide context specific guidance in software development. For example, if the development context such as a large project size or outsourcing requirements demands that the Plan-driven method be adopted, then the JCM can offer ways of improving team member morale through manipulation of the five characteristics. Skill variety in a Taylorist development project can be enhanced through job rotation by involving developers in requirement gathering activities as well as quality assurance. A shared vision for a Taylorist project can help increase task identity and significance for the individual team members. It will enable the team members see the big picture (task identity) and make them realize the importance (task significance) of the specialized tasks which they perform. This in turn will increase the meaningfulness of work and sustain high team member morale in line with predictions of the JCM and supporting empirical evidences.

These recommendations through the application of the JCM in the above examples also happen to be the software development best practices as in CMMi (Capability Maturity Model – Integrated).

The guidance emanating from the JCM is thus well aligned with known best practices of software development, generating added confidence in the relevance of this model in the context of software development.

Conclusion and Contribution

For teams to be effective, its members should find their work motivating (Hackman and Oldham, 1980). Employee motivation is considered the single most important antecedent of software quality and employee productivity (Boehm, 1981; McConnell, 1998). Yet, managing employee motivation continues to be problematic (Procaccino and Verner, 2006). Keeping this context in view, we use work design theories, perspectives and prescriptions in understanding SDMs and their role in engendering motivation of software engineers.

The results of the study show that independent of the methods used for software development the 5 job characteristics in general have beneficial impacts on motivational and work outcomes (see Table 10 below for a summary).

These findings are useful for software development organizations. They suggest that enriching the job content by manipulating the job characteristics can be used to improve motivation of the employees. A need has been expressed in literature for tailoring methods of software development for improved outcomes (Fitzgerald, Hartnett, and Conboy 2006). Improving motivation has significant impact on satisfaction, work exhaustion and performance of team members. These findings suggest that software development organizations can control psycho-social and work outcomes of software development by managing the job characteristics.

Additionally, although there is a huge body of research on SDMs yet it is the practitioners and consultants that have largely driven the creation and
dissemination of these methods (Abrahamsson, Conboy and Wang, 2009). This approach has propelled the practice of software development ahead of formalized theory resulting in a plethora of new methodologies (Ågerfalk, Fitzgerald and Slaughter, 2009). However, reinventions and remixes of existing practices have limited value (Dingsøyr, Nerur, Balijepally and Moe, 2012). As Jacobson and Spence (2009) point out, sound theoretical roots help us glean the essential concepts, or the “truths” of software development that are methodology-independent. Applying the JCM to SDMs provides these methodology-independent truths of software development by suggesting that work design characteristics maybe at the core of team member motivation and performance of software development projects.

<table>
<thead>
<tr>
<th>Job Characteristics</th>
<th>Self-Determined Motivation</th>
<th>Non-Self-Determined Motivation</th>
<th>Satisfaction</th>
<th>Work Exhaustion</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill Variety</td>
<td>Positive</td>
<td>Positive</td>
<td>Positive</td>
<td>Negative</td>
<td>Not significant</td>
</tr>
<tr>
<td>Task Identity</td>
<td>Positive</td>
<td>Positive</td>
<td>Not significant</td>
<td>Negative</td>
<td>Not significant</td>
</tr>
<tr>
<td>Task Significance</td>
<td>Positive</td>
<td>Positive</td>
<td>Positive</td>
<td>Negative</td>
<td>Positive</td>
</tr>
<tr>
<td>Autonomy</td>
<td>Positive</td>
<td>Positive</td>
<td>Positive</td>
<td>Negative</td>
<td>Positive</td>
</tr>
<tr>
<td>Feedback</td>
<td>Positive</td>
<td>Positive</td>
<td>Positive</td>
<td>Not significant</td>
<td>Positive</td>
</tr>
</tbody>
</table>

Broy (2011) argued that, “engineering disciplines must be based on scientific practices and theory to justify their approaches and to give scientific evidence for why and where their methods work properly”. This study and its finding can be considered an attempt in that direction. To elaborate further, we can conceptualize a software development method as composed of practices. Also, although not explicitly stated, the practices themselves may be based on one or more of the job characteristics of the JCM. For example the success of Agile practices such as paired programming is based on and can be explained by the rapid feedback it provides to the developers. The benefits of developing working products in rapid iterative cycles can be explained by the enhanced significance of the tasks completed (through completion of “whole” rather than narrow tasks) and getting early feedback from the users of the product. In addition developing whole, meaningful and working products makes it easy for developers to identify with the tasks.

The practice and benefit of having a customer representative on the Agile development team can be explained by the feedback they can provide to the developers. Software is more intangible than physical products. Coupled with rapidly evolving customer requirements frequent feedback is necessary to provide information about the extent to which the individual and team efforts meet the expectations of customers. Additionally, when teams provide its members contact with beneficiaries (such as customers), team members understand the beneficiary perspective thereby enhancing the significance of the tasks they perform (Parker and Axtell, 2001). The resulting empathy and awareness of the impact of their work on beneficiaries strengthen intrinsic motivation of the
employees resulting in higher levels of effort, persistence, and helping behavior (Grant, 2007).

Thus using a single theory, the JCM, one can get insights into why certain software development practices work and the way to improve them through enhancement of the job characteristics. Much of research in software engineering research is guided by what is termed as “theory fragments” (Stol and Fitzgerald, 2013). Although theoretical investigations have been conducted before to assess and improve specific practices such as unit testing, code releases, refactoring, communication (Pikkarainen, Haikara, Salo, Abrahamsson. and Still, 2008; Petersen and Wohlin, 2010; Müller and Höfer, 2007; Sharp and Robinson, 2004) a case has been made for a general theory of software development methods (Ralph, Johnson, and Jordan, 2013). This study can be considered a nascent effort in that direction.

Another contribution of the study is identifying the distinguishing characteristics of Agile and Plan-driven methods of software development from perspective of job characteristics. “Theoretically comprehending the distinction between Agile methods and Plan-driven methods is a concern begging for research attention.” (Dingsøyr, Nerur, Balijepally and Moe, 2012). The results of the study show that Agile methods perform better on all the 5 job characteristics and that the 5 job characteristics might therefore represent indicators of agility. Skill variety represents the flexibility dimension of agility by creating reserve capacity (Mumford, 2006). Autonomy enables quick response to changing conditions. Under uncertain conditions the Taylorist practice of managers planning, assigning and controlling tasks of software engineers may not work.

Further increasing task identity and significance through assignment of “meaningful work” comprising whole rather than narrow tasks develops team member knowledge of the product and the process as a whole. This knowledge helps in development of larger analytical and problem-solving capacities critical in responding quickly under uncertain conditions (Niepce and Molleman, 1998). Finally, timely feedback from customers, team members and managers help make timely changes, thereby reducing rework and cycle times. In line with the JCM, it is these differences that account for and explain the higher motivation and satisfaction of team members of Agile projects compared to team members of projects using Plan driven methods observed across multiple studies (Cockburn and Highsmith, 2001; Layman, Williams and Cunningham, 2004; Mannaro, Melis and Marchesi, 2004; Mann and Maurer, 2005; Melnik and Maurer, 2006; Dyba and Dingsøyr, 2008).

Finally, the results of the study indicate that using JDS software development organizations can estimate and manage the job characteristics of their software development projects. At high levels of job characteristics are the archetypical Agile SDMs and at low levels are the archetypical Plan-driven SDMs. In between these levels are the tailoring possibilities for creating hybrid methods. For example, the methodology independent impacts of job characteristics can provide a mechanism for enhancing job characteristics of Plan-driven methods and thereby facilitate in “balancing scalability with agility” (Cao, Mohan, Xu, and Ramesh, 2009).

Limitations and Future research

However these contributions should be viewed in light of the following limitations. Although software development methods are broadly classified into two categories, the Agile methods and the Plan-driven or Taylorist methods, within each category there are many different methods each with their own principles and practices making comparisons between them confusing. For
example, the many Agile methods currently in use such as Extreme programming, Scrum, Crystal methodologies, Dynamic Software development method (DSDM), Feature Driven Development (FDD) and Lean Software Development Method (LSDM) each focus heavily on some of the principles of the agile manifesto and completely ignoring others making it impossible to reach any conclusions on specific agile methods and their use (Conboy and Fitzgerald, 2004). The sample size did not permit further statistical analyses of differences within these two major paradigms.

Hence the results only broadly reflect the distinction between Agile methods and plan-driven methods. However, it may be noted that agile and plan-driven methods are “ideal types”. Ideal types help in theorizing and promote deeper understanding of a phenomenon (Jessop, 2002). But software development methods are never found in pure form. Many variants of the pure forms exist. Mixes and remixes of practices (Dingsøyr, Nerur, Balijepally and Moe, 2012) promoted by these two major paradigms are found in work situations depending on context such as organizational culture, volatility of user requirements, skill profile of the development team and complexity of the software developed. Yet, the findings of the study do provide useful insights into the characteristic differences between the two major paradigms of software development. Future research may design studies to assess specific methods of software development to evaluate their strengths and weaknesses from a work design perspective. It will help project managers to decide on which particular method to adopt for their software development projects.

References


Motivating Potential of Software Development Methods / Kakar


Franca, C., Sharp, H. and Da Silva, F. (2014). Motivated software engineers are engaged and concentrated while satisfied ones are happy, in Proceedings of ESEM Sept, Torino Italy.


Appendix A: Job Characteristics Items from JDS (Hackman and Oldham, 1974)

1. In general, how significant or important is the job you do?
   1 = Not at all significant; the outcome of the work are not likely to affect anyone in any important way
   9 = Highly significant; the outcome of the work can affect other team members in very important ways

2. To what extent do managers or coworkers let you know how well you are doing on your job?
   1 = Very little, people almost never let me know how well I am doing on my job
   9 = Very much; managers or coworkers provide me with almost constant “feedback” about how well I am doing on my job

3. To what extent does the job permit you to decide on your own how to go about doing the work?
   1 = Very little; the job gives a person almost no personal “say” about how and when the work is done
   9 = Very much, the job gives the person almost complete responsibility for deciding how and when the work is done

4. To what extent does the job involve doing a “whole” and identifiable piece of work?
   1 = The job is only a tiny part of the overall piece of work; the results of the person’s activities cannot be seen in the final product or service
   9 = The job involves doing the whole piece of work from start to finish; the results of the person’s activities are easily seen in the final product or service

5. How much variety is there in your job?
   1 = Very little; the job requires the person to do routine things over and over again
   9 = Very much; the job requires the person to do many different things using a number of different skills and talents
## Appendix B: Measures Used in the Study

<table>
<thead>
<tr>
<th>Measures and Items</th>
<th>Notes: Intrinsic motivation (IM) <em>4,8,15</em>; integrated regulation (IR) <em>5,10,18</em>; identified regulation (IDR) <em>1,7,14</em>; introjected regulation (IR) <em>6,11,13</em>; external regulation (ER) <em>2,9,16</em>; amotivation (AM) <em>3,12,17.</em></th>
</tr>
</thead>
</table>
| **Motivation** (Tremblay, Blanchard, Taylor, Pelletier and Villeneuve, 2009) | 1. Because this is the type of work I have chosen to do to attain a certain lifestyle.  
2. For the income it will provide me.  
3. I ask myself this question, I don't seem to be able to manage the important tasks related to this work.  
4. Because I derive much pleasure from learning new things.  
5. Because it has become a fundamental part of who I am.  
6. Because I want to succeed at this job, if not I would be very ashamed of myself.  
7. Because I chose this type of work to attain my career goals.  
8. For the satisfaction I experience from taking on interesting challenges.  
9. Because it will allow me to earn money.  
10. Because it is part of the way in which I have chosen to live my life.  
11. Because I want to be very good at this work, otherwise I would be very disappointed.  
12. I don't know why, we are provided with unrealistic working conditions.  
13. Because I want to be a “winner” in life.  
14. Because it is the type of work I have chosen to attain certain important objectives.  
15. For the satisfaction I experience when I am successful at doing difficult tasks.  
16. Because this type of work will provide me with security.  
17. I don’t know, too much is expected of us.  
18. Because this kind of job is a part of my life. |

**Satisfaction** (McKnight, Philips and Hardgrave, 2009)  
- Generally speaking, I am very satisfied with the project.  
- I frequently thought of quitting this project [Reverse Coded]  
- I am satisfied with the kind of work I did on the project  

**Work Exhaustion** (McKnight, Philips and Hardgrave, 2009)  
- I feel emotionally drained from my work.  
- I feel used up at the end of the work day.  
- I feel fatigued when I get up in the morning and have to face another day on the job.  
- I feel burned out from my work.  

**Performance**  
- The team member completed all work assigned to him on time  
- The team member consistently delivered work of high quality  
- The team member demonstrated high productivity
Appendix C: Results of Factor Analyses

<table>
<thead>
<tr>
<th>Items</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>1. IDR</td>
<td>0.866</td>
<td>-0.003</td>
<td>0.051</td>
<td>0.013</td>
<td>-0.041</td>
<td>0.120</td>
<td>0.001</td>
<td>-0.039</td>
<td>0.046</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>2. ER</td>
<td>-0.044</td>
<td>0.844</td>
<td>-0.049</td>
<td>-0.039</td>
<td>-0.054</td>
<td>0.040</td>
<td>-0.047</td>
<td>-0.047</td>
<td>0.037</td>
<td>0.050</td>
<td></td>
</tr>
<tr>
<td>3. AM</td>
<td>-0.042</td>
<td>0.072</td>
<td>0.866</td>
<td>-0.062</td>
<td>0.032</td>
<td>0.043</td>
<td>-0.107</td>
<td>-0.025</td>
<td>0.018</td>
<td>-0.052</td>
<td></td>
</tr>
<tr>
<td>4. IM</td>
<td>-0.053</td>
<td>-0.061</td>
<td>-0.050</td>
<td>0.769</td>
<td>0.003</td>
<td>0.288</td>
<td>0.059</td>
<td>0.061</td>
<td>-0.109</td>
<td>0.042</td>
<td></td>
</tr>
<tr>
<td>5. IR</td>
<td>-0.037</td>
<td>-0.024</td>
<td>-0.024</td>
<td>-0.013</td>
<td>0.859</td>
<td>0.027</td>
<td>0.074</td>
<td>-0.022</td>
<td>0.097</td>
<td>-0.048</td>
<td></td>
</tr>
<tr>
<td>6. INR</td>
<td>-0.043</td>
<td>-0.018</td>
<td>-0.078</td>
<td>-0.030</td>
<td>0.048</td>
<td>0.871</td>
<td>0.080</td>
<td>0.215</td>
<td>0.084</td>
<td>-0.060</td>
<td></td>
</tr>
<tr>
<td>7. IDR</td>
<td>0.759</td>
<td>-0.034</td>
<td>-0.056</td>
<td>-0.077</td>
<td>0.114</td>
<td>0.007</td>
<td>0.077</td>
<td>0.034</td>
<td>0.258</td>
<td>0.055</td>
<td></td>
</tr>
<tr>
<td>8. IM</td>
<td>-0.017</td>
<td>-0.019</td>
<td>-0.031</td>
<td>0.840</td>
<td>0.103</td>
<td>0.042</td>
<td>0.150</td>
<td>0.117</td>
<td>0.094</td>
<td>-0.018</td>
<td></td>
</tr>
<tr>
<td>9. ER</td>
<td>-0.059</td>
<td>0.640</td>
<td>-0.080</td>
<td>-0.007</td>
<td>-0.144</td>
<td>0.137</td>
<td>0.034</td>
<td>0.251</td>
<td>0.145</td>
<td>0.037</td>
<td></td>
</tr>
<tr>
<td>10. IR</td>
<td>-0.067</td>
<td>-0.052</td>
<td>-0.041</td>
<td>-0.043</td>
<td>0.739</td>
<td>0.139</td>
<td>0.076</td>
<td>0.010</td>
<td>0.180</td>
<td>-0.027</td>
<td></td>
</tr>
<tr>
<td>11. INR</td>
<td>-0.084</td>
<td>-0.065</td>
<td>-0.040</td>
<td>-0.002</td>
<td>0.028</td>
<td>0.919</td>
<td>0.019</td>
<td>0.035</td>
<td>0.017</td>
<td>-0.054</td>
<td></td>
</tr>
<tr>
<td>12. AM</td>
<td>-0.029</td>
<td>-0.031</td>
<td>0.776</td>
<td>-0.028</td>
<td>-0.026</td>
<td>0.101</td>
<td>0.077</td>
<td>0.074</td>
<td>0.075</td>
<td>0.041</td>
<td></td>
</tr>
<tr>
<td>13. IR</td>
<td>-0.042</td>
<td>-0.061</td>
<td>-0.030</td>
<td>-0.014</td>
<td>0.044</td>
<td>0.889</td>
<td>0.012</td>
<td>0.084</td>
<td>0.050</td>
<td>-0.046</td>
<td></td>
</tr>
<tr>
<td>14. IDR</td>
<td>0.855</td>
<td>-0.048</td>
<td>-0.032</td>
<td>-0.032</td>
<td>-0.043</td>
<td>0.113</td>
<td>-0.024</td>
<td>0.050</td>
<td>0.066</td>
<td>-0.052</td>
<td></td>
</tr>
<tr>
<td>15. IM</td>
<td>-0.047</td>
<td>-0.028</td>
<td>-0.025</td>
<td>0.932</td>
<td>0.009</td>
<td>0.065</td>
<td>-0.019</td>
<td>0.057</td>
<td>-0.035</td>
<td>-0.062</td>
<td></td>
</tr>
<tr>
<td>16. ER</td>
<td>-0.021</td>
<td>0.715</td>
<td>-0.022</td>
<td>-0.070</td>
<td>0.124</td>
<td>-0.070</td>
<td>-0.047</td>
<td>-0.082</td>
<td>-0.036</td>
<td>-0.059</td>
<td></td>
</tr>
<tr>
<td>17. AM</td>
<td>-0.051</td>
<td>-0.028</td>
<td>0.818</td>
<td>-0.041</td>
<td>0.317</td>
<td>0.020</td>
<td>0.044</td>
<td>-0.014</td>
<td>0.048</td>
<td>-0.051</td>
<td></td>
</tr>
<tr>
<td>18. IR</td>
<td>-0.028</td>
<td>-0.046</td>
<td>-0.074</td>
<td>-0.067</td>
<td>0.832</td>
<td>0.112</td>
<td>0.078</td>
<td>-0.002</td>
<td>-0.050</td>
<td>-0.047</td>
<td></td>
</tr>
</tbody>
</table>

Notes: Intrinsic motivation(IM): 4, 8, 15; integrated regulation(IR): 5, 10, 18; identified regulation (IDR): 1, 7, 14;

About the Author

Adarsh Kumar Karak is a Ph.D in Management Science (MIS track) with an interest in investigating software development processes and practices. He has over 3 decades of experience in the software industry and has worked as consultant for many Fortune 500 companies including British Airways and Saudi American Bank. He is currently working as an Assistant Professor in the department of Computer Information Systems at Alabama State University. His work has been published in journals such as Information and Software Technology, Computers in Human Behavior, Interacting with Computers, International Journal on Human-Computer Studies, Journal of Computer Information Systems, Interacting with Computers, Information Systems Management, AIS Transactions on Human Computer Interactions and ACM Transactions of Management Information Systems.