Innovation on Digital Platforms: Impacts of Control Portfolios on Novelty

Alan Hevner*
University of South Florida, Tampa, FL USA

Onkar Malgonde†
Northern Illinois University, DeKalb, IL USA

Abstract

The development of software applications on digital platforms requires both agility and restraint to meet rapidly changing user requirements while adding novel features to a platform-based application domain. User value creation focuses on exploring the solution space to innovate and attract new customers while retaining existing customers. In this pilot study, we analyze the essential tensions between software project controls and the development activities to achieve novelty in the software product. Drawing from cognitive theories of creativity and reasoning, we posit the need for both informal controls that enhance creativity and formal controls that enhance reasoning in a balanced portfolio of project controls. Two case studies provide preliminary evidence that a well-balanced portfolio of controls can result in the effective design and implementation of novel product features. We position the case studies in the context of digital platforms to bound our definitions of control mechanisms and novelty. We conclude with implications for software development on digital platforms and future research directions.

Key Words: Control, novelty, creativity, reasoning, digital platforms.

1 Introduction

Recent advancements in software systems and information technologies are driving digital transformation initiatives within organizations and renewed organizational focus on innovation [14, 24, 26]. To support such endeavors, we are witnessing significant changes in business practices such as partner networks, subscription-based usage, and open innovation. With a renewed focus on innovation in digital era, software development projects are increasingly identifying and incorporating enabling technologies and tools such as platform-based application development, low-code development platforms, enterprise application packages, and prototyping tools [24]. Many organizations focus on achieving value creation opportunities in the context of digital platforms which represent a mainstream channel for development and deployment of software development projects. Software application development on digital platforms requires project teams to achieve application-platform match, realize application-market match, exceed core value proposition of the platform, and provide novelty of the application. This is in addition to traditional outcomes of project success such as project efficiency, quality, and adaptiveness [11, 19, 23].

A key research question is how to achieve the right balance between project controls while supporting the creative design and implementation of novel features. Prior research has found contradictory results. Several studies focusing on the control of software projects identify a negative effect of control mechanisms on innovation outcomes [4, 6] suggesting a "stifling of creativity and limiting of adaptability" [6, p. 225]. However, many software projects with portfolios of existing control mechanisms do effectively release novel products on different platforms and in many application domains.

We explore the activities needed to produce a novel application. Drawing from research on human cognition, novelty is achieved via synergy between creative, divergent thinking and reasoning, convergent thinking [12, 21]. Divergent thinking engenders imagination, provocation, unstructured syntheses, serendipitous discovery, and answers that break with conformity. This mode of cognition focuses on the synthetic generation of multiple disparate answers to a given problem [1]. Convergent thinking refers to the mode of human cognition that strives for the generation of a single, concrete, accurate, and effective solution. Thus, divergent cognition (creativity) produces many possible new and interesting solutions; while convergent cognition (reasoning) assesses the feasibility of these solutions and identifies the best solution for implementation. Thus, we explore the following research question:

> *How does a portfolio of formal and informal controls relate to the creative and reasoning activities required for the development of novel features in software projects on digital platforms?*

To address this question, we propose a research model and assess it on two real-world case studies. Via qualitative interviews with members of the two software development projects, we identify the control mechanisms applied and the

* School of Information Systems and Management. Muma College of Business. Email: ahevner@usf.edu.
† Operations Management and Information Systems. College of Business. Email: omalgonde@niu.edu.

development activities performed to achieve novel application features. We organize this paper as follows. In sections 2 and 3, respectively, brief literature surveys present the grounding theories of control and novelty. Section 4 presents a proposed research model relating controls and novelty. In Section 5, we discuss our research methodology, data collection, and analysis approach. We present the criteria for selecting two case studies. Section 6 discusses the results of each case study. We conclude in section 7 with the implications of this research for building control portfolios to support the design and implementation of novel features on digital platforms. Future research directions are presented.

2 Software Development Controls

Controls in software projects have received considerable attention in research and practice [19]. However, several surveys identify a lack of rigorous study on the effect of control mechanisms on novelty of software application features [15, 19]. Control theory as applied to software projects recognizes two categories of controls – formal and informal [6].

2.1 Formal Controls

Formal control types are classified as input control, behavior control, outcome control, and emergent outcome control. The key distinction for formal control is the presence of an identifiable controller and controlee. Each type of formal control is briefly described.

- **Input Control**: The controller assigns resources (inputs) to the development project team (controlee) that are sufficient to successfully complete a desired result. Input controls are closely related to the Theory of Effectuation in the field of entrepreneurship [22]. An entrepreneurial software team is provided with means to achieve project aspirations [15]. The team decides how best to affect the desired results. There has been limited discussion on the use of input control for software projects [23]. However, in the context of R&D projects in the pharmaceutical industry, Cardinal [4] finds empirical evidence in support of input controls leading to incremental and radical innovation. Software project managers can alter resources such as team composition, technical environments, tools, and knowledge resources, among others, for the project team to facilitate identification and assimilation of novel features for the focal application.
- **Behavior Control**: The controller uses processes and rules to direct controlees towards accomplishment of organizational goals. In software project control, behavior control is exercised using mandated routines such as meetings and development methodologies that signal use of specific methods in the project. Prior research attributes use of behavior control to outcomes of project adaptiveness, efficiency, and quality [23]. In the context of R&D projects in the pharmaceutical industry, Cardinal [4] finds a negative effect of behavior controls on novelty. Some studies posit a negative effect of behavior control due to rigidity and lack of experimentation which stems from overly specifying controlee behaviors [6, 11].
- **Outcome Control**: The controller specifies an outcome and evaluates the project based on the controlee achieving the outcome. For outcome control to be effective, the controller should be able to specify the outcome a priori and measure the achieved outcome. Typically, software requirement/specifications and fixed timelines form popular control mechanisms for outcome control. An important challenge with the use of outcome control is the ability of the controller to define a novel product at the start of a project. In the digital platform environment, novelty of an application will change over time, as platform and competitors update their offerings. Also, measuring novelty of an application is challenging [15].
- **Emergent Outcome Controls**: Instead of novel outcomes being predictable at the beginning of a project, novel outcomes often emerge during the course of a software development project. The controller sets defined project milestones to assess the trajectory of the emerging novel product. The use of scope boundaries and ongoing feedback are important forms of emergent outcome controls that allow the project team to revise outcomes that are difficult to identify a priori by facilitating feedback [13]. Scope boundaries channel the team's efforts while allowing autonomy within the boundaries. For development teams developing applications for digital platforms, emergent outcome controls provide mechanisms which can enable the team to explore the technological space provided by the platform and seek feedback from within and outside the development team.

2.2 Informal Controls

Informal controls rely on a software team's shared values and vision of the application. There is no strict hierarchy of team structure between controller and controlee. Two forms of informal controls are clan and self.

- **Clan Control**: Shared values and goals among team members motivate the project to a successful result. Chua et al. [5] find that clan controls need to be developed over time with careful maneuvering to be effective. Experienced teams demonstrate higher levels of clan control. Although difficult to implement [9], clan controls demand minimal monitoring once implemented. Empirical evidence suggests a positive effect of clan control on project's success [5, 23]. In digital platform environment, clan control can play an important role in channeling the team's efforts towards developing a novel application [15].
- **Self-Control**: Team members have internal motivations to self-direct their actions to achieve project goals. Prior research suggests a positive effect of self-control on project's success [14]. In the dynamic environment of digital platforms, it is important to enable individual autonomy in order to identify and design novel features that

will set apart the focal application. The project manager may identify appropriate control mechanisms to enable team members to exercise self-controls to experiment with features and technological advancements to develop novel features for the application.

3 Software Application Novelty

Traditionally, organizations develop innovative product lines through a linear value chain [20]; products are designed, developed, and marketed by a single firm. However, with pervasive digital innovations and technology, the locus of organizational innovation has shifted to digital software platforms which rely on external entities to develop innovative solutions. Thus, current conceptualizations of novelty in an application refer to the features and extensions offered by the application relative to the platform and other competing applications.

For our research, we extend this definition of novelty, as the dependent variable of our study, to include content provided by the application, data sources and their designs, user interfaces, alerts/messages, and platform's ecosystem that distinguishes the focal application from its competition (competing applications that may or may not be on the same platform). Consequently, novelty of the application is not limited to its features. Novelty for a focal application may arise from its choice of platform since the application's user may not differentiate[1] between the application and its platform.

To achieve application novelty on digital platforms, the software development team must effectively iterate between two cognitive modes – *creative* activities that generate new ideas and *reasoning* activities that analyze the feasibility of the new ideas to determine how best to implement the novel application features. The following subsections briefly survey and distinguish these two essential mindsets in software development.

3.1 Creativity in Software Development

The literature on the cognitive bases of creativity is fragmented with little consensus about the neural mechanisms underlying creativity. This is true for the literature on creativity as a whole and for the sub-domains of divergent thinking, aesthetics (e.g., style, art, music), and insight. Creativity is viewed as a complex computational model of activities 'in' many areas of the brain [10]. Conceptualizing and treating creativity as if it is a single entity fails to accommodate its complexity and infers that it comprises a limited number of fundamental processes and brain structures underlying it. Dietrich and Kanso [8] point out that this is likely to be a fallacy, and that "it is hard to believe that creative behavior in all its manifestations – from carrying out exquisitely choreographed dance moves, to scientific discovery, constructing poems and coming up with ingenious ideas of what to do with a brick - engages a common set of brain areas or depends on a limited set of mental processes" (p. 845).

While neuroscience provides no definitive answers on the origin of creativity, the software engineering community has applied several development processes that aim to generate novel artifacts. Creative processes incorporate the dynamics of the (socio-cognitive) activities underlying an artifact's complexity, creation, composition, and later use and evolution. High levels of creativity are fostered by radical, out-of-the-box thinking and non-conventional approaches for the development of new ideas. Organization policies that foster creativity are key; particularly those that provide the entrepreneurial team time to think and try out their own ideas. Specific techniques that could be used include *genius grants* and *bootlegging* [7, 17]. Other examples are *tinkering time* and *hack-a-thons*.

Seminal research on creative teams by Amabile and Pillemer [2] identifies the following four components as integral to the creative process:

- Domain-relevant skills include intelligence, expertise, knowledge, technical skills, and talent in the particular domain in which the team is working;
- Creativity-relevant processes are enabled by personality and cognitive characteristics that lend themselves to taking new perspectives on problems, such as independence, risk taking, self-discipline in generating ideas, and a tolerance for ambiguity.
- Intrinsic task motivation is seen as a central tenet. People are most creative when they feel motivated primarily by the interest, enjoyment, satisfaction and challenge of the work itself – and not by extrinsic motivators.
- The social environment, the only external component, addresses the working conditions that support creative activity. Negative organizational settings harshly criticize new ideas, emphasize political problems, stress the status quo, impose excessive time pressures, and support low-risk attitudes of top management. While positive organizational settings stimulate creativity with clear and compelling management visions, work teams with diverse skills working collaboratively, freedom to investigate ideas, and mechanisms for developing new ideas and norms of sharing ideas.

It is important to note that Amabile's work is based on two important assumptions. First, there is a continuum from relatively low, everyday levels of adaptive creativity to the higher levels of creativity found in significant inventions and scientific discoveries. Second, there are degrees of creativity exhibited in the work of any single individual at different points of time and circumstances [2].

3.2 Reasoning in Software Development

A student once asked Linus Pauling, "Dr. Pauling, how does one go about having good ideas?" He replied, "You have lots of ideas and throw away the bad ones." [2, p. 116]. Effective innovation requires more than just the generation of many

[1] In enterprise grade applications, users are often unaware about the digital platform and its offerings when using the application.

creative ideas. Many creative individuals waste time, energy, and resources chasing infeasible and unprofitable hunches into blind alleys. Successful innovation also requires the intellectual control to refine creative thinking into practical solutions. Such control is dependent on the cognitive skills of reasoning and judgment.

Human cognitive reasoning reflects thinking in which plans are made, hypotheses are formed, and conclusions are drawn on the basis of evidence in the form of data, past experience, or knowledge. While creativity often calls for divergent thinking to break out of mindsets; reasoning calls for convergent thinking to refine ideas into practical artifacts and actions. Moving design ideas from 'blue sky' to artifact instantiations requires goal setting and a plan to answer the following types of systems development questions:

- *Is the design feasible?* - Can the proposed design be implemented and does the proposed design meet the requirements of the stakeholders and the platform?
- *Does the design have value?* - Does the design offer benefits unmatched by competing candidate designs? Here the objective becomes to establish an ordinal valuation that can be used to rank candidate designs.
- *How can the design be most effectively represented?* – How can we best communicate the intricacies of the design to collaborators, implementers (e.g., architects, programmers), and other stakeholders?
- *How best to construct the actual use artifacts?* How do we guide the construction of the use artifact? As examples - a blueprint is a construction artifact that serves to guide the physical construction of a house; source code is a construction artifact that serves to generate the programs that are distributed to users.

Closely related to reason is the human cognitive facility to judge, or evaluate, ideas at various design stages of the development process. The goal of judgment is to predict the future; to predict which candidate designs will be better than others. Without the ability to narrow the field (i.e., design space) it would be impossible to refine many good ideas down to one 'satisfactory' design artifact. This is a very tricky area of human cognition since it involves self-criticism, self-esteem, and motivation. However, studies have shown that humans are capable of making effective and rapid judgments based on first impressions (e.g. [3]). Beyond first impressions, measurements and evaluations are based on the rigorous definition of utility functions that estimate the values of candidate designs in order to facilitate the ranking of alternatives.

4 Research Model

While the topics of software development controls and software application novelty have received considerable attention in the research literature individually [19], there exists few formal studies of how these topics are related. Thus, grounded by the previous two sections, we propose the following research model (Figure 1) for our study.

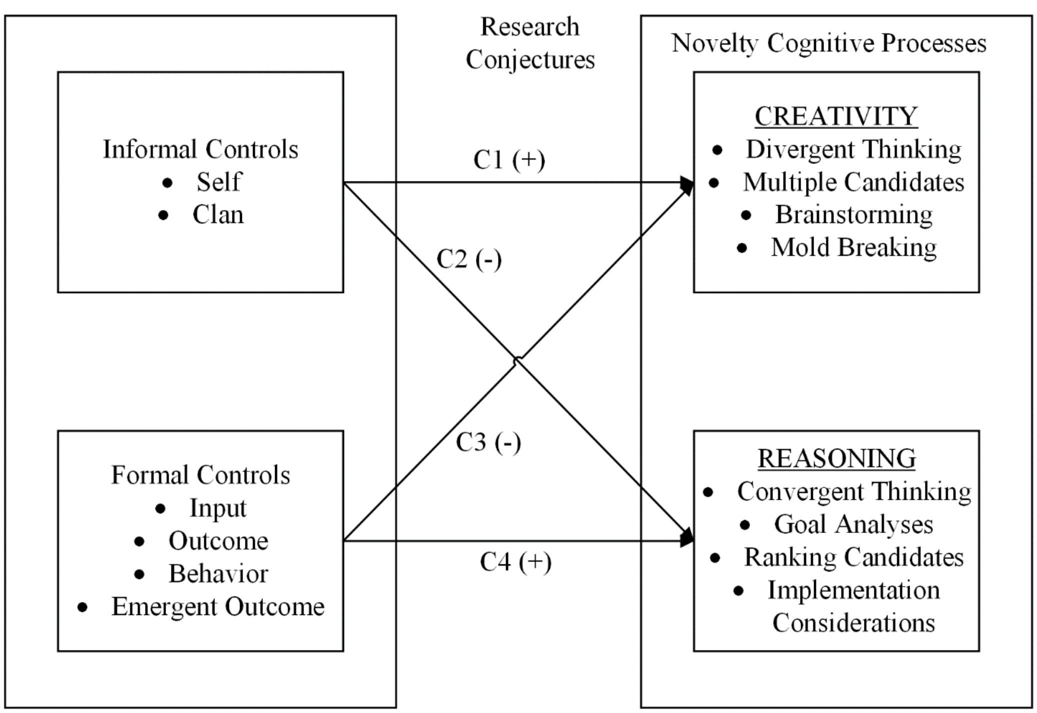

Figure 1: The relationship of controls and novelty cognitive processes

Four research conjectures are illustrated in the model:

Conjecture 1: Informal controls increase support for creativity in the design and development of novel application features on digital platforms.

Conjecture 2: Informal controls decrease support for reasoning in the design and development of novel application features on digital platforms.

Conjecture 3: Formal controls decrease support for creativity in the design and development of novel application features on digital platforms.

Conjecture 4: Formal controls increase support for reasoning in the design and development of novel application features on digital platforms.

Guided by these research conjectures, we perform a study to investigate the relationships of software controls and application novelty in software development projects on digital networks.

5 Research Methodology

This research studies the relationship of control mechanisms and the use of cognitive processes to produce novelty in platform-based applications. We conduct case studies to answer our research question. A case study methodology is appropriate when 'how' questions are posed in the research [25]. Case studies also allow us to extract a nuanced understanding of the control mechanisms identified by software development teams that contribute to novelty of the application. The unit of analysis is the project team that is developing the application on digital platform. To provide empirical grounding, we conduct two case studies.

5.1 Case Selection and Site Description

Selection of projects is driven by the following criteria: (a) the application is built on a digital platform, (b) stakeholder roles (controllers and controlees) can be identified, (c) competing applications for the focal application exist, (d) novel features in the focal application are identifiable, and (e) ownership of application and platform are not held by same organization(s). Table 1 provides details about the case study sites, number of interviews at each site, informant roles in the interviews, and brief descriptions of the projects at the case study sites.

5.2 Data Collections

To test the efficacy of our selection criteria, interview protocols, and theoretical understandings, we performed two pilots [16]. Our first pilot location is an IT-department of a large public university in the Southeastern United States. The application under consideration allows universities to complete their reporting obligations for state-mandated requirements. The second pilot location is a Fortune-500 organization with a large development team of analysts, architects, and a project manager. This team is developing an application that supports online subscription of enterprise software. The application connects to multiple external platforms, increasing the complexity of the project. The pilots refined our protocol and data analysis methods.

The questioning protocol forms the basis for each interview with occasional deviations to accommodate any contemporary issues such as reordering questions based on an interviewee's response or dropping certain questions that are not consistent with an interviewee's role in the project. Follow up questions (not included in the protocol) may be included to seek clarification and/or reconfirmation. Finally, questions exploring interviewee's role in the development project may be included to better understand the controls adopted by the team. Table 1 provides a summary of the two case study sites, named AT and TB. In case of AT, the majority of interviews took place on-site whereas a small fraction are individual online sessions. In case of TB, all the 7 interviews are individual online sessions.

Table 1: Summary of case study sites

Organization	Interviews	Informant Roles	Project Description
AT is a software consulting firm specializing in development, maintenance, and deployment of software applications across different industries.	9	• Delivery Lead • Team Lead • Senior Developers • UI Designer • UI Developer • Technical Architect	The client (a non-profit organization) wants to develop a mobile application (iOS-based) that would allow healthcare professionals to stream on-demand educational content, videos, support dynamic notetaking, and resume playback.
TB develops IT solutions to challenges in different domains such as CRM, Healthcare, and Operations	7	• Product Owner • Product Manager • Practice Manager • Sales Consultant • Technical Architect • Solution Architect • Functional Consultant	The product is a cloud-based Healthcare management application and competes with other offerings on Microsoft Azure platform. TB partners with select customers (hospitals) to develop features which are incorporated into the product – streamline patient care with CRM platform and consolidate patient care.

5.3 Data Analysis

To answer our research questions, we analyzed the qualitative data in two phases [18]. First, open coding systematically labeled our data to identify events, actions, and interactions. Second, axial coding related categories and subcategories around the data. We used independent coders with experience in software development projects to code our data. Coders were trained using the pilot interviews' transcripts. We discussed and clarified any ambiguity in conceptual understanding and operational definitions. Coders were blind to the research question. We performed within case analysis, followed by cross-case analysis. Our analysis focused on identifying key findings on the relationships between control mechanisms and the novel features of the software system under development.

6 Case Study Results

The two case studies supplied rich detail for qualitative analyses of the software development projects in the two organizations. We present the results of interviews with a focus on the control processes employed and the novel features produced in each software development project.

6.1 Case Study 1: AT

6.1.1Control Mechanisms. We find use of different technical environments to facilitate experimentation with new ideas. Specifically, teams use *sandbox* environments to trial new ideas, *demo* environments to integrate new ideas with existing application, and *quality assurance* environments to test new ideas with existing application features; in addition to the *production* environment that hosts the actual application. Setting, maintaining, and transferring artifacts from such environments utilizes resources (time and cost). In case of AT, we find controls in the form of changes in team composition to facilitate development of novel features. Specifically, the video playback feature with positional saving and scrolling for AT's application is not supported by the platform's native capabilities and requires technical expertise. We find use of *collaborative sessions* aided by interactive mockups and designs to be a key behavioral control mechanism to identify and refine novel features for the application. We find different labels for these sessions: backlog refining/grooming, or brainstorming sessions (including a session with the project manager of a Fortune 500 organization).

We find support in the use of ongoing feedback during development of the application. The traditional conceptualization of ongoing feedback has a directionality from clients/users to the development team. However, we find that feedback may be bidirectional during and/or after iterations.

Table 2 summarizes formal control mechanisms in AT that contribute to the novelty of the application.

Regarding informal controls, we find evidence for the use of clan control in the AT project. Discussions with the client's liaisons pertaining to requirements are typically viewed as meetings that facilitate the project's understanding of the domain. However, as part of these discussions, the team discusses alternative ways to either improve known requirements or recommend new requirements based on the team's prior experience. In case of AT, we see evidence of a clan mentality where entrepreneurial thinking [15] is encouraged so that the client is successful. However, we also see the negative influence of a "consultant" mentality. Specifically, AT's consultants realize that their role is limited to a module. Similarly, AT's leadership team acknowledges that decisions on features are made by the client based on time, cost, and desirability of recommended feature. Such client-vendor relationship may hinder identification and assimilation of novel features as AT may be wary to discuss potential features. The presence of experienced software developers on the AT project provides some evidence for individual self-controls.

Table 3 summarizes informal control mechanisms for AT project.

6.1.2 Novel Features. Novel features in AT project are threefold. First, the application allows its users to seamlessly stream content such as text, pictures, and videos on mobile devices. This requires dynamic adjustments to the content. The content is stored on the client's servers. Previously the native video playback feature from the platform lacked finesse. Second, the application enables users to make and retrieve notes while they are watching videos. Finally, the user can resume video playback from the last viewed location. These features set the focal application apart from competing applications on the platform and off-platform alternatives including a client's website. Some of these features exist in other domains. For example, resuming video playback from the last view position has been staple for video streaming application on the same platform. However, the ability to stream video content and extend platform's native capability is novel in the client's competition space. Table 4 lists the novel features of the AT application.

6.2 Case Study 2: TB

6.2.1 Control Mechanisms. At TB, the management team identified *springboard* clients ("early adopters" as noted by TB's Product Manager) that provided market needs and differentiators in exchange for access to the application. We classify partnerships with such clients as input control due to the emphasis by controller to leverage access to springboard clients and temporary association with such clients. One of TB's team members (title—Solution Consultant) is a registered nurse (the target users of TB's application) who participates in product demonstrations and identification and vetting of potential features. Inclusion of a team member who can provide users' perspective is another input control mechanism. TB teams use mock-up screens and designs so that all stakeholders can visualize potential features, alter designs to visualize focal feature, and identify approaches to incorporate potential features in the application. In addition to discussions based on interactive mockups.

TB's team also performs configurational changes to the

Table 2: Formal control mechanisms in AT project

Control Mode	Control Mechanisms
Input Controls	• Change team composition (add and drop skills/personnel) to identify and develop novel features • Setup different technical environments to execute proof of concepts and integrate ideas in the application
Behavior Controls	• Facilitate workshops with users/clients at intervals • Interactions/feedback loops during iteration demo • Collaborative discussions/workshops between team and users with user interface mockups • Use technical capabilities to identify new features • Discuss technical approaches to achieve new features
Output Controls	No evidence found
Emergent Outcome Controls	• Feedback during and after each iteration • Ongoing feedback from application's usage data • Content and how it is served by the application

Table 3: Informal control mechanisms in AT project

Control Mode	Control Mechanisms
Clan Controls	• Shared understanding on success criteria • Prefer physical presence for client meetings whenever possible
Self-Controls	• Experienced software developers

Table 4: Novel features of AT application

Novel Features
• Allow users to seamlessly stream content such as text, images, and videos, on mobile devices. This requires dynamic adjustments to the content. The content is stored on the client's servers. The native video playback feature from the platform lacked finesse.
• Application enables users to make and retrieve notes while they are watching videos.
• User can resume video playback from the last viewed location.

existing application and platform, where possible, to visualize new ideas for the application. We find identification of technology and tools to be made upfront which determines the scope boundaries for technical exploration. In case of TB, there exist organizational restrictions on the use of technology and tools provided by a vendor. We also find that choice of the digital platform introduces a major scope boundary for the team as technical capabilities and tools are bounded. The project team must identify alternatives that can be supported by resources within the scope boundary. We also find use of feedback mechanisms, part of emergent outcome control mode, to facilitate an individual's exploration of specific ideas. Table 5 summarizes formal control mechanisms in TB project.

In the case of TB, we find a shared understanding and importance of novelty across team members. The shared success criteria encourage the team to identify and vet alternatives. Also, we find that team members appreciate and recognize the value contributed by other team members. For example, analysts recognize the possibilities and limitations faced by the technical team in implementing proposed features. To address limitations, analysts identify feasible alternatives and discuss with the technical team. Further, we find common consensus on the importance of certain processes and events. For example, meetings and discussions with potential customers is an important opportunity to verify and gather new feature ideas. For self-controls, proactive team members may experiment to identify novel features for the application. After implementation, typically as a proof-of-concept, other team members with closer market knowledge may adapt and integrate the novel feature. Given the shared understanding and importance on identification of novel features, the team is willing to discuss and improve any ideas put forth. Table 6 summarizes informal control mechanisms in TB project.

6.2.2 Novel Features. Novel features for TB's application are threefold. First, data management in the application is patient-centric whereas competitors use an event-centric approach. With a patient-centric approach, application's users can view all records for a patient on the dashboard. Second, user experience is highly rated. This includes the application's ease of use and performance. Third, easy integration with Microsoft's productivity suite that may be already functional at client's location. Another area of novelty for TB's team is the choice of platform. Microsoft's Azure platform integrates with Microsoft's productivity suite such as calendar, business intelligence reports, emails, and so on, allowing TB's application to differentiate itself from its competitors that use different platform.

Table 5: Formal control mechanisms in TB project

Control Mode	Control Mechanisms
Input Controls	• Identify and partner with springboard clients • Choice of focal platform—platform's features, ecosystem, and maturity—to distinguish the application from competition • Application's user as part of the team • Technical members attend seminars and conferences hosted by the platform • Setup different technical environments to execute proof of concepts and integrate ideas in the application
Behavior Controls	• Configurational changes to platform before discussing features • Interactions/feedback loops during experimentation, testing, and documentation (within team) • Interactions/feedback loops with springboard clients • Collaborative discussions/workshops between team and users with user interface mockups • Use technical capabilities to identify new features • Discuss technical approaches to achieve new features
Output Controls	No evidence found
Emergent Outcome Controls	• Application is always ready for demo and feedback • Ongoing feedback from application's usage data • Content and how it is served by the application

Table 6: Informal control mechanisms in TB project

Control Mode	Control Mechanisms
Clan Control	• Shared understanding on the importance of processes (for example, team visit to springboard client's site) to identify novel features • Challenge team members to extend existing feature set
Self-Control	• Proactive team members try new ideas and discuss alternatives

Table 7: Novel features of TB application

Novel Features
• Data management in the application is patient-centric whereas competitors use an event-centric approach. With a patient-centric approach, application's users can view all records for a patient on the dashboard.
• User experience is highly rated. This includes the application's ease of use and performance.
• Easy integration with Microsoft's productivity suite that may be already functional at client's location.

6.3 AT and TB case study findings

Summarizing AT and TB case study findings, we find compelling use of both formal and informal control mechanisms that lead to identification and assimilation of novel features which exceed the platform's core proposition and/or differentiate the application from its competition. Specifically, we find use of a mixed control portfolio consisting of formal and informal controls. We do not find support for traditional outcome control mechanisms which we attribute to challenges in specifying novel outcomes a priori in the dynamic digital platform environment. Finally, we find positive influence of a more long-term orientation of the team (TB case) in comparison to a short-term focus (AT case) which can be attributed to the perpetual mode of application under development in platform environments.

We identified well-defined novel features in the platform-based applications for each case study. In our interviews with the development team, we elicited the paths of divergent and convergent thinking that led to the novel features. Based on how these paths were influenced by formal and informal controls, we interpret these qualitative data and examine each of the four research conjectures in our research model.

6.3.1 Informal Controls and Creativity. We find convincing evidence for role of informal controls in increasing support for creativity in design and development of novel application features. For example, in case of AT, the novel feature of resuming video playback is introduced via individual creativity via self-controls. Specifically, challenges with platform's technology did not lend itself to develop this feature. A senior programmer in AT's team ran an experiment. This experimentation involves identifying multiple approaches (brainstorming) that can address the technological challenge (divergent thinking). In case of AT and TB, development teams create interactable mock-ups to facilitate discussions related to application's features. Discussions based on these mock-ups help the team to identify new features and/or alter existing features (brainstorm) and identify new process flows (mold-breaking).

Clan control mechanisms such as shared understanding of the importance of novel features for the application, acknowledgement of challenges involved in identifying novel features, and a collective mindset to facilitate identification of novel features. In case of TB, proactive team members identify potential features that can extend existing features for the application. Together, informal control mechanisms help to identify multiple candidates (novel features) for the application.

To identify multiple candidates, informal controls facilitate a culture which emphasizes divergent thinking and builds tolerance for radical ideas.

6.3.2 Informal Control and Reasoning. We find evidence for the role of informal controls in decreasing support for reasoning in design and development of novel application features. For example, in case of TB, novel features related to user experience do not focus on implementation considerations (reasoning) at the onset. Instead, the team focuses on identifying multiple designs (creativity) without considerations of feasibility and implementation details, such as what can be supported by the platform's capability or the need to build new modules. In case of AT's novel feature of dynamic content display on different devices, AT's team considers divergent thinking (creativity) rather than convergent thinking (reasoning) as the team seeks to identify possible usage scenarios on devices of different size, capability, and software environment.

Following our earlier discussion on information controls and creativity, informal controls do not lend themselves to rank candidates, converge multiple ideas, or implementation considerations. Rather, informal controls seek to extend existing thinking and ideas to identify multiple candidates without considerations to implementation, feasibility, and priority. Reasoning-based processes may be time dependent in comparison to creativity promoted by informal controls. Whereas creative processes seek to identify novel features without any constraints of implementation, reasoning processes may be limited by time. For example, technological limitations may hinder a novel idea. In summary, informal controls increasingly support creativity whereas decrease support for reasoning.

6.3.3 Formal Controls and Creativity. We find evidence for the role of formal controls in decreasing support for creativity in design and development of novel application features. This conjecture reflects the long-standing concerns that inflexible software development processes constrain creativity. For example, in case of AT, a new team member was added (input control) to implement the novel video playback feature. This new role was specifically required to implement a given requirement within predefined implementation boundaries. In case of TB, discussion (behavior control) on application's features focuses on identifying implementable features (reasoning) rather than deriving a list of new options (creativity).

Formal control mechanisms focus on the means to accomplish the ideas identified by the team. Formal controls do not facilitate identification of multiple candidates. Although, some behavioral control mechanisms may facilitate brainstorming sessions for the team, these are often limited to discussions on feasibility based on cost and time. Formal control mechanisms aim to converge the development process such that specified deliverables are accomplished. Such a focus does not lend to divergent thinking and identification of multiple candidates.

6.3.4 Formal Controls and Reasoning. We find evidence for the role of formal controls in increasing support for reasoning in design and development of novel application features. In case of TB, teams use feedback from customers and application usage data (emergent outcome control) to prioritize (reasoning) novel user interface features in the application. In case of AT, a time constraint introduced by an external source such as an application demo in a conference, invokes use of an emergent outcome control and behavior control to rank and converge (reasoning) novel feature alternatives, so that they can be demoed in the conference.

Formal control mechanisms facilitate prioritization of novel features identified in earlier iterations. Prioritization is often based on criteria such as deliverables, technical feasibility, impacts on existing application, tool support, available time, and costs. These processes are goal-driven that require evidence-based ranking of alternatives. For example, AT and TB teams use emergent outcome control mechanisms to specify technological boundaries and seek feedback on intermediate outcomes. These mechanisms are focusing on converging the broader novel feature that can be experienced in the application.

7 Discussion and Future Research Directions

In this research, we perform two rigorous case studies on platform-based application development projects in order to identify control mechanisms and novel application features on digital platforms. Based on analyses of the case studies, we find that informal and formal controls both contribute to platform innovations; however, in different cognitive ways. Further, we contribute to theory by presenting four conjectures for further study: (a) informal controls support increasing creativity, (b) informal controls support decreasing reasoning, (c) formal controls support decreasing creativity, and (d) formal controls support increasing reasoning.

In our case study research, we find empirical evidence in support of platform's role to enable both formal and informal controls. The role of a project manager to build a mixed portfolio of controls to maximize team members' contributions to application novelty is a key finding of this research. Platforms are a common point of reference during discussions, decision-making, and collaborative sessions. For example, as teams identify new features, a major focus is how to implement the potential features using the platform's current capabilities. The digital platform also facilitates high control communication and evaluation congruence. The platform plays the role of an anchor that is referenced to alter the control portfolio as the project evolves and to evaluate the current appropriateness of the portfolio. As new and/or updated platform offerings are visible to all, antecedents to control portfolio changes are visible and more likely to be accepted by the team.

In an organizational context, other control modes such as structure, market, and culture, have been considered in related domains [4]. However, software systems project control has been theorized to focus on "temporary organizations" that require different control activities [23] than the larger counterparts of organizational control. In our analysis, we find evidence that challenges this notion of controlling a temporary organization. Applications developed on digital platforms may be perpetually in the state of development due to changes in market and platform.

Our study has two major implications for software project research. First, our findings align with related domains where control's effectiveness to introduce innovation has been established [4]. This finding calls for deeper studies of how controls and novelty relate in software development projects [6]. Second, this research addresses the recent call to investigate project controls in the digital era [24]. Specifically, we isolate the relationships of controls and novelty and perform a preliminary, qualitative study. Such a purpose-oriented focus allows us to investigate the required balance between value-appropriation concerns and value-creation requirements in the digital era.

We present four conjectures on control modes and novelty cognitive processes of creativity and reasoning. While additional empirical research is required to formally hypothesize and test these conjectures, we believe this research sets the stage for an increased understanding of the importance of incorporating cognitive processes in software engineering literature. Consideration of novelty cognitive processes is particularly important for software engineering because of the increasing importance of digital innovation [14]. Also, cognitive processes seek to focus individuals and teams to control portfolios that are more effective in software development projects. This research can complement future studies which incorporate organizational and team innovation literature in software engineering.

There are several important limitations and future research directions in our study. First, we consider only two small development projects with in-house applications. Future research can consider other project settings such as offshoring, large project teams, different application domains, and so on. Second, our findings are limited to projects where novelty is incremental. As we move in the digital era, one of the major challenges for future research on the impacts of control to novelty is to study projects that focus on radical innovation. Third, we did not explore the effect of platform's type on the project control—AT's platform caters to consumers whereas TB's platform caters to enterprises. Consequently, AT's platform has tight integration with products and services offered by the digital platform whereas TB's platform has tight integration with other platform-based services provided by the owners. The platform's coupling has consequences for the project as platforms feature updates, releases, and technology that are dependent on other services.

References

[1] T. Amabile, *How to Kill Creativity*, Harvard Business School Publishing, Vol. 87, 1998.

[2] T. M. Amabile. and J. Pillemer, "Perspectives on the Social Psychology of Creativity," *The Journal of Creative Behavior*, 46(1):3-15, 2012.

[3] N. Ambady and R. Rosenthal, "Half a Minute: Predicting Teacher Evaluations from Thin Slices of Nonverbal Behavior and Physical Attractiveness," *Journal of Personality and Social Psychology*, 64(3):431, 1993.

[4] L. B. Cardinal, "Technological Innovation in the Pharmaceutical Industry: The Use of Organizational Control in Managing Research and Development," *Organization Science*, 12(1):19-36, 200

[5] C. E. H. Chua, W. K. Lim, C. Soh, and S. K. Sia, "Enacting Clan Control in Complex IT Projects: A Social Capital Perspective," *MIS Quarterly*, 36(2):577-600, 2012.

[6] W. A. Cram, K. Brohman, and R. B. Gallupe, "Information Systems Control: A Review and Framework for Emerging Information Systems Processes," *Journal of Association for Information Systems*, 17(4):216-266, 2016.

[7] P. Criscuolo, A. Salter, and A. L. Ter Wal, "Going Underground: Bootlegging and Individual Innovative Performance," *Organization Science*, 25(5):1287-1305, 2014.

[8] A. Dietrich and R. Kanso, "A Review of EEG, ERP, and Neuroimaging Studies of Creativity and Insight," *Psychological Bulletin*, 136(5):822, 2010.

[9] K. M. Eisenhardt, "Control: Organizational and Economic Approaches," *Management Science*, 31(2):134-149, 1985.

[10] A. Fink, M. Benedek, R. H. Grabner, B. Staudt, and A. C. Neubauer, "Creativity Meets Neuroscience: Experimental Tasks for the Neuroscientific Study of Creative Thinking," *Methods*, 42(1):68-76, 2007.

[11] B. Fitzgerald, "Formalized Systems Development Methodologies: A Critical Perspective," *Information Systems Journal*, 6(1):3-23, 1996.

[12] J. P. Guilford, "Creativity: Yesterday, Today and Tomorrow," *The Journal of Creative Behavior*, 1(1):3-14, 1967.

[13] M. L. Harris, R. W. Collins, and A. R. Hevner, "Control of Flexible Software Development Under Uncertainty," *Information Systems Research*, 20(3):400-419, 2009.

[14] A. Hevner, and S. Gregor, "Envisioning Entrepreneurship and Digital Innovation through a Design Science Research Lens: A Matrix Approach," Information & Management, 2020.

[15] A. Hevner and O. Malgonde, "Effectual Application Development on Digital Platform,". *Electronic Markets*, 29(3):407-421, 2019.

[16] O. Malgonde, *An Effectual Approach for the Development of Novel Applications on Digital Platforms*, University of South Florida, 2018.

[17] Y. Masoudnia and M. Szwejczewski, "Bootlegging in the R&D Departments of High-Technology Firms," *Research-Technology Management*, 55(5):35-42, 2012.

[18] M. Miles, A. M. Huberman, and J. Saldana, *Qualitative Data Analysis: A Methods Sourcebook*, SAGE Publications, 2013.

[19] W. J. Orlikowski, "Integrated information environment or matrix of control? The contradictory implications of

information technology," *Accounting, Management and Information Technologies*, 1(1):9-42, 1991.

[20] G. G. Parker, M. W. Van Alstyne, and S. P. Choudary, *Platform Revolution*, W. W. Norton & Company, 2016.

[21] M. A. Runco, *Creativity: Theories and Themes: Research, Development, and Practice*, Elsevier 2014.

[22] S. Sarasvathy, "Causation and Effectuation: Toward a Theoretical Shift from Economic Inevitability to Entrepreneurial Contingency," *The Academy of Management Review*, 26(2):243-263, 2001.

[23] M. Wiener, M. Mahring, U. Remus, and C. Saunders, "Control Configuration and Control Enactment in Information Systems Projects: Review and Expanded Theoretical Framework," *MIS Quarterly*, 40(3):741-774, 2016.

[24] M. Wiener, M. Mähring, U. Remus, C. Saunders, and W. A. Cram, "Moving IS Project Control Research into the Digital Era: The 'Why' of Control and the Concept of Control Purpose," *Information Systems Research*, 30(4):1387-1401, 2019.

[25] R. Yin, *Case Study Research: Design and Methods*, SAGE Publications, 2008.

[26] Y. Yoo, O. Henfridsson, and K. Lyytinen, "The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research*, 21(4):724-735, 2010.

Alan R. Hevner is a Distinguished University Professor and Eminent Scholar in the Information Systems and Decision Sciences Department in the Muma College of Business at the University of South Florida. He holds the Citigroup/Hidden River Chair of Distributed Technology. Dr. Hevner's areas of research interest includes design science research, digital innovation, information systems development, software engineering, distributed database systems, and healthcare systems. He has published over 250 research papers on these topics and has consulted for a number of Fortune 500 companies. Dr. Hevner received a Ph.D. in Computer Science from Purdue University. He has held faculty positions at the University of Maryland and the University of Minnesota. Dr. Hevner is a Fellow of the American Association for the Advancement of Science (AAAS), a Fellow of the Association for Information Systems (AIS), and a Fellow of IEEE. He is a member of ACM and INFORMS. Additional honors include selection as a Parnas Fellow at Lero, the Irish software research center, a Schoeller Senior Fellow at Friedrich Alexander University in Germany, and the 2018 Distinguished Alumnus award from the Purdue University Computer Science Department. From 2006 to 2009, he served as a program manager at the U.S. National Science Foundation (NSF) in the Computer and Information Science and Engineering (CISE) Directorate.

Onkar S. Malgonde is an Assistant Professor at the Operations Management & Information Systems, College of Business, Northern Illinois University. He received a B. Engg in Information Technology in 2010 from University of Pune and a PhD in MIS from the University of South Florida in 2018. His research interests include software engineering, machine learning, and digital platforms. His research has been published in MIS Quarterly, Empirical Software Engineering, Electronic Markets, Workshop on Information Systems and Technology (WITS), International Conference on Design Science Research in Information Systems and Technology (DESRIST), and Americas Conference of Information Systems (AMCIS).