7-2018

Towards a dynamic theory of enacted complexity

Kenneth T. GOH
Singapore Management University, kennethgoh@smu.edu.sg

Brian T. PENTLAND

TOWARDS A DYNAMIC THEORY OF ENACTED COMPLEXITY

ABSTRACT

To develop new theory about the dynamics of enacted ask complexity, we analyze 15-months of field data from a video game development project consisting of observations, interviews, and an archival analysis of 2,428 tasks to present a novel way of conceptualizing and visualizing the complexity of emergent processual phenomena.

Keywords: Complexity, dynamic model, interdependence, narrative network

Task complexity is not something that can be directly observed. Instead, researchers have constructed indicators of complexity that are based on an idealized description of the task that is separate from the enactment of the task (Campbell, 1988; Wood, 1986). For experimental research on well-scripted, individual tasks, it makes sense to treat the task as separate from the task-doer (Hackman, 1969). For emergent, collective tasks, like video game development (Cohendet & Simon, 2016) or organizational strategy (Jarzabkowski, 2004), this separation is hard to justify because the tasks to be performed are not easily determined beforehand – goals and means-ends relationship may both be ambiguous. The involvement of multiple, interdependent actors also has implications on performance. To illustrate this point, contrast the case of two actors performing a task to the case when three actors are involved. In the second case, the third actor creates additional coordination needs but also allows the triad to perform more tasks in parallel. Thus, for emergent tasks that involve multiple actors, separating the task from the task-doer partially obscures certain aspects of complexity. For these kinds of phenomena, we need new ways of seeing and theorizing about task complexity.

In this paper, we address the question: **how can seeing the complexity of emergent processual phenomena inform our theorizing about the dynamics of enacted complexity?** By *processual phenomena*, we mean any organizing process (Weick, 1979) that unfolds over time (Abbott, 2016). This includes a broad spectrum of phenomena, at many different time scales, from organizational routines (Feldman et al, 2016) to creative projects (Obstfeld, 2012) to organizational strategy (Jarzabkowski, 2004). By *emergent*, we mean that because processual phenomena unfold over time, they are inherently and continually "becoming" (Tsoukas & Chia, 2002). We use the metaphor of "seeing" to stand for the combination of theory, method, and insight that signifies progress in any empirical science.

We start with a brief review of conventional ways of seeing complexity in tasks, processes, projects and other processual phenomena. These conventional views assume that the task or process exists as an independent entity, separate from the social and material context (Hackman, 1969). The conventional view provides the basis for what we call "descriptive" measures of complexity. We then offer an alternative perspective that is aligned with so-called "strong" process theory (Hernes, 2014; Langley & Tsoukas, 2017; Tsoukas & Chia, 2002). This perspective matches the emergent, processual phenomena, and provides a better basis for a dynamic theory of complexity. To motivate and illustrate this theory, we use data from field research on video game development to analyze enacted complexity over the course of a game development project. We use this case to demonstrate a new way of seeing and theorizing about the dynamics of enacted complexity. We then conclude the paper with a discussion of how this new way of seeing enables new directions in organizational research.

THEORY

Our perspective flows from the ontological shift that forms the basis for processual organizational research (Langley & Tsoukas, 2017). The philosophical roots of this perspective can be traced to Whitehead, James, Mead, and more recently the work of Weick, Rescher, Hernes, Chia, and others. The basic insight is simple. As Weick (1979: 95) observed, "organizations are grounded in interlocked behaviors rather than interlocked people." Putting actions at the center, rather than actors, transforms the way that we see organizational phenomena. Pentland et al (2017) describe this change in perspective as a "Copernican revolution". Placing the Sun at the center of the solar system (rather than the Earth) required the invention of new ways of seeing (i.e., the telescope). The transformative potential of the action-centric, processual revolution requires new ways of seeing, as well.

To grasp the significance of this transformation for the analysis of task complexity, consider an analogy between tasks and the wind.[1] In the conventional view, wind can be treated as an independent entity that has properties like velocity and direction. The wind exists even if its velocity is zero. In the processual view, wind only exists when it is blowing. Similarly, from a processual perspective, a task only exists when it is being carried out (Hærem, Pentland, & Miller, 2015). This aligns with the view that the social world is a continually unfolding process (Tsoukas & Chia, 2002) and thus the "dynamic, unfolding process becomes the primary unit of analysis rather than the constituent elements themselves" (Emirbayer & Mische, 1998, p. 287). In the paragraphs that follow, we elaborate on the contrast between these two perspectives.

Conventional View of Task Complexity

The conventional view of complexity in organizational research derives from Wood (1986) and Campbell (1988). Wood (1986) identified three basic dimensions of task complexity:

- **Component complexity** refers to the number of distinct, nonredundant acts required to complete a task. A task with more required acts is more complex.
- **Coordinative complexity** refers to the number of precedence relationships between the actions that convert task inputs into task products. Longer sequences of dependencies indicate a more complex task.

---

[1] This example is borrowed from Mesle and Dibben (2016), who borrowed it from a keynote address by Martha Feldman, who borrowed it from Mustafa Emirbayer (1997), who borrowed it from Norbert Elias.

- **Dynamic complexity** refers to changes in the other two dimensions of complexity at two or more distinct points in time.

To this list, Campbell (1988) added multiple pathways to task completion, multiple outcomes, uncertain means-ends relations, and others, but these dimensions have rarely been operationalized in empirical research. In their review of over 700 studies citing either Wood (1986) or Campbell (1988), Hærem and colleagues (2015) found only one study (Banker, Davis, & Slaughter, 1998) that operationalized all three dimensions defined by Wood (1986). In practice, the most common approach is to rely exclusively on the components -- the "required acts" -- as an indicator of complexity. This indicator makes intuitive sense and it is convenient to measure.

However, the conventional view of task complexity has a number of limitations, especially when applied to emergent tasks that are carried out by multiple actors. Following Alvesson and Sandberg (2011), Hærem et al (2015) examine the assumptions of the conventional view. Using the same basic ingredients as Wood (1986) (i.e., required acts, information cues and outputs), they reformulate the concept of task complexity in a way that aligns with a processual ontology. It extends to tasks performed by multiple actors, integrates the social and material context of task performance, and most important for our purposes, it provides an index of task complexity based on empirical observations of task performance. For this reason, it provides the basis for developing and testing theory about complexity in field research. Hærem et al (2015) argue that the conventional view tends to limit the applicability of task complexity in areas where the task itself is an emergent phenomenon with multiple participants.

An Alternative View of Complexity

The reformulated concept of task complexity is based on the number of paths through a network of actions that represents a task (Hærem et al., 2015). This kind of network is also referred to as a narrative network (Pentland & Feldman, 2007; Pentland et al., 2017). Rather than counting discrete acts, it counts the *paths*, which are the *sequential combinations* of acts that are observed during performance of the task. More paths indicate a more complex task. The network is a directed graph where the nodes represent distinct acts (as in Wood, 1986) and the edges represent sequential relations between those acts. Pentland and Liu (2017) describe methods for constructing action networks from data collected in field research. These networks can be automatically constructed from event logs or observations using software provided by Pentland and colleagues (2015, 2016).

The number of paths is affected by the number of acts as well as the density of the connections. Hærem et al (2015) show that in limiting cases (e.g. one actor), their definition of complexity matches Woods' definition (1986). However, where Wood (1986) and Campbell (1988) posit a linear relationship between required acts and complexity, Hærem et al (2015) argue that the relationship should be exponential. In other domains, complexity is typically an exponential function (Moldoveanu & Bauer, 2004). Thus, the network-based view is more in keeping with contemporary understanding of complexity.

Another key difference concerns the definition of the network. In a radical departure from the conventional view, Hærem et al (2015) include the actor in the definition of each action. It matters who does what. This happens whenever there is a division of labor among different roles (e.g., in performing a surgery, doctors and nurses perform different parts of the overall task). Thus, Hærem et al (2015) incorporate the basic concept of role, defined as the set of actions that can be carried out by an actor (Blau & Scott, 1962). In a single actor task, or in any task where each actor performs a distinct set of actions, this makes no difference. But in situations where the same actions can potentially be performed by multiple actors, the number of actors can influence complexity dramatically. This conforms to the well-established principle in software development that that adding people can make a project more complex (Brooks, 1974).

This way of theorizing task complexity has two important advantages for our purposes here. First, it uses readily observable behavior as the basis for operationalizing enacted complexity. This helps make complexity visible. Visualization is especially valuable for processual phenomena (Feldman, 2016). While we are getting better at conceptualizing processual phenomena (Langley & Tsoukas, 2017), our tools for seeing these phenomena are weak. Feldman (2016) argues that current approaches to representing process (figures with double headed arrows between abstract categories) do just as much to obscure the underlying phenomenon as to reveal it. Textual descriptions, while they can be "rich" in some respects, are impoverished by the inherent limitations of grammar and the linearity of text (Mesle & Dibben, 2016). A visual representation that is based on specific empirical evidence provides a new way of seeing processual phenomena.

Second, defining complexity as a function of network structure suggests that complexity will increase (or decrease) as nodes and edges are added (or removed) from the network of actions that represents a task. To construct a dynamic theory of task complexity, we need to identify mechanisms that add (or remove) nodes and edges from the network. Note that this can happen throughout the performance of a task or a project, when activities are added (or removed), and when members join (or leave) a project team, and so forth. We describe these mechanisms in our analysis of data from the video game development project.

Adding dynamics to the model is important because Hærem et al (2015), like Wood (1986), are still basically describing snapshots of a process. They provide a novel way to conceptualize complexity, but do not provide a theory of how it changes. The network representation provides a simple way to conceptualize and operationalize changes that affect complexity: (a) adding/removing nodes and (b) adding/removed edges. These mechanisms align with Wood (1986): nodes indicate component complexity and edges indicate coordinative complexity. But the nodes and edges are defined differently, so that they include the social and material context, and the overall complexity is indexed by the number of paths in the resulting graph.

Compared to the traditional view, the action network provides a more holistic representation of processual phenomena "becoming" (Tsoukas & Chia, 2002) as performed by multiple, interdependent actors. It provides a way to complement the existing process studies that have predominantly been qualitative (Langley, Smallman, Tsoukas, & Van de Ven, 2013). By mapping action networks, we can complement the richness and depth of qualitative approaches with an objective measure and comparison of enacted complexity.

METHODS

Research Setting

The setting for our study is ProjectBQ at a videogame development studio, GameSG (both pseudonyms), based in a mid-Atlantic city in the United States. During the period of data collection, GameSG was a 10-year old studio that employed approximately 60 employees, mostly under 30 years of age, with expertise in software engineering, game design, and technical art. Prior development projects at GameSG included games on various platforms (e.g., mobile phones, stand-alone entertainment systems, TV plug-in games, internet browser games) for a wide spectrum of clients that included video game publishers, media conglomerates, theme parks, and a startup toy company.

Project teams in GameSG were usually composed of members with expertise in one of the following skill sets - game design, software engineering, technical art, script writing, animation, sound composition, and project management. The composition of team members in ProjectBQ was typical in this regard. The team was led by a core group of functional "leads" consisting of the producer, a lead designer, a technical lead, and an art lead. Each lead was responsible for coordinating work in that functional domain and acting as a gatekeeper for the quality of work produced. Project leads were also directly involved on high level decisions about the design and functionality of the game. The producer managed deadlines, the pace of work, and access to resources for the team. They played a boundary-spanning role between the team and other stakeholders such as GameSG management, other project teams, and the client. The team size for ProjectBQ ranged from 8 to 15 over a 14-month period.

ProjectBQ involved developing a single-player, action-adventure mobile game in which players help the leader of a mouse tribe resist the corrupting influence of its enemies. In this game, players completed missions in various virtual worlds by battling against enemies in turn-based combat. The game was funded by a non-profit with the goal of promoting anti-drug messages through unstructured learning methods. The project was a "serious" game intended to teach teenagers about responding to uncomfortable situations (e.g., substance abuse, risky behavior) without succumbing to peer pressure.

We picked videogame development as an exemplary setting for studying enacted complexity because it is a collective task that is ambiguous and emergent: there are an endless number of possibilities for combining elements to create a game. Videogames are an interactive virtual experience produced by a computer program onto a display device that people engage in for entertainment. Although games are also used in more "serious" settings such as education and training simulations, there is always an element of interactivity and engagement with the player. However, how this interactivity and engagement manifests in the context of the game is rarely obvious at the outset of game development (Cohendet and Simon, 2016).

These characteristics of videogame development can be considered a type of creative project (Obtsfeld, 2012). Creative projects consist of an emergent trajectory of interdependent action initiated and orchestrated by multiple actors to introduce change into a social context. The nature of these departures could be in the form of new elements, or new linkages between familiar elements. The trajectory of action required to create the videogame does not follow a set plan because outcomes are ambiguous which means that tasks and actions to be performed are emergent.

Creative projects are an ideal setting for studying enacted complexity because these projects are a source of emergent actions enacted by multiple actors who are "projecting a new end stage" (p. 1572). Since "repetition is not a guide on what to do next" (p. 1571), actors are less constrained by past routines and have a considerable degree of agency over their actions. Creative projects thus allow for more endogenous variation in enacted complexity independent of descriptive complexity.

Data Collection

Our research design incorporated data from non-participant observation, interviews, and archival materials. Data was collected over 15 months as part of a longer two-year study on the routines in video game development.

Between May 2011 to August 2012, the first author was a non-participant observer on ProjectBQ. These observations included team meetings (n = 39), client meetings (n = 7), and play test sessions (n = 4). Team meetings included daily fifteen-minute "scrum" meetings (n = 29) where team members met to schedule and coordinate their tasks for the day, retrospective meetings where they reviewed work processes (n = 2), and general discussions about the project (n = 8). During these meetings, notes were taken about the purpose of the meeting, what was said and by whom, and the author's impressions of what transpired during the meeting.

In addition to data from observations, both ad hoc informal (n = 11) and formal semi-structured interviews (n = 5) were conducted with team members. The informal interviews focused on getting status updates on the project while formal semi-structured interviews were about 60 minutes long and focused on gaining an in-depth understanding of specific episodes during the project.

The third source of data were archival materials such as task schedules, planning documents, meeting notes, and budgets. The primary document we relied on to construct networks of action patterns were archives of task schedules that contained logs of tasks assigned to each individual. These documents were updated daily by the team and daily versions of these documents were downloaded between May 2011 and February 2012 (n = 122). As an archival source, the task schedules are particularly suitable for capturing chronologies of actions over long periods of time (Langley et al., 2013).

The scrum sheets were used to create a database of tasks, the "story" or goal that it meant to accomplish, the actors associated with the tasks, and when the task started and ended. A "difference report" was created for each day by comparing scrum sheets with the most recent version to identify which tasks were added or removed, and the progress made on the task. From these daily difference reports, a list of actions was created (n = 2,803). Starting and ending dates for each task were also extracted from the difference reports. Tasks without a start and end date were removed as these tasks were not acted upon, resulting in final list of 2,428 tasks. These tasks were then grouped by stories and sequenced according to the following order: 1) when the task ended, 2) when it was started, 3) the order in which the task was added to the database. The last criterion was necessary to determine the ordering of tasks that shared similar start dates and end dates.

Data Analysis

In keeping with our goal of seeing and theorizing about complexity as it was enacted over the course of the project, we analyzed data chronologically, as a narrative. The data analysis consisted of three main activities: (1) constructing a project narrative that provides a context for understanding and explaining specific actions that affect complexity; (2) constructing a series of action networks that allow us to measure the enacted complexity throughout the project; and (3) using the project narrative to interpret and theorize about changes in enacted complexity.

Constructing the project narrative. We began by constructing a timeline of events from interviews with informants. These interviews were professionally transcribed and analyzed using nVivo software to identify periods, major events, and the critical actors associated with the temporal unfolding of the project (Langley, 1999). We drew on the first author's observations of the project team to validate our timeline of the project. Each observational event was dated and summarized. We then compared the events provided by informants with these observations to validate the timeline.

To create a more detailed narrative, we augmented the basic project timeline by iterating between the interviews and the observations with an emphasis on the contextual circumstances surrounding interpretations of why events occurred, individual thoughts and feelings in response to actors and incidents, and histories. This narrative provided a depth of insight into temporal unfolding of project that extended temporally across the past and into the future, and across actors that included individuals, the team, and external stakeholders.

Measuring enacted complexity. As described by Hærem et al (2015), measuring enacted complexity consists of two main steps: constructing the action network and counting the paths. Constructing the action network requires coding the actions and roles involved in carrying out the project. Pentland and Liu (2017) provide a detailed description of how to construct action networks from a variety of different data sources.

Coding actions. We used a constant comparative process (Glaser & Strauss, 1967) to develop task categories database of tasks collected from the scrum sheets. Categories were developed by iterating between the first author's familiarity with the context, field notes, and other archival documents to understand the intent of the task.

This process involved forming initial clusters tasks to minimize differences within clusters while maximizing differences between clusters. An initial set of categories were then developed from these clusters. New tasks were then compared with earlier tasks in the same category. If a newly categorized task appeared to be different from other tasks in the same category, this would be reconciled by attempting to refine the definitions and properties of these categories to accommodate the new data. This process of constantly comparing new data with existing codes was continued until a level of stability was reached. From twelve initial categories, the list was ultimately reduced to the following six categories: Administration, Experimenting, Building, Revision, Refinement, and Testing (Table 1). Figure 1 shows the distribution of these categories across Sprints.

INSERT FIGURE 1 ABOUT HERE

Coding roles. In a similar manner, the primary actor responsible for each task in the database was categorized into an organizational role by the first author based on his familiarity with the research setting. These roles were Design, Art, Tech, and Analytics (Figure 2). Together with the actions, these roles define the possible actions in the action network. The four roles and six task categories meant that there were potentially 24 unique role-task pairs.

INSERT FIGURE 2 ABOUT HERE

Bracketing the sprints. The BQ project was implemented with an agile project methodology (Moe, Dingsøyr, & Dybå, 2010; Sutherland & Sutherland, 2014), which meant that the project was broken into three-week long phases called "sprints." For our analysis of enacted complexity, we temporally bracketed the data (Langley, 1999) into three-week windows that corresponded with the dates for each sprint.

Visualizing the action network and computing enacted complexity for each sprint. The coded sequences were entered into a simple spreadsheet (.XLS format) for analysis using *ThreadNet* (Pentland et al., 2015, 2016). *ThreadNet* calculated indices for enacted complexity and created graphs of action network for the project in aggregate (Figure 3) and for each sprint (Figure 4). This provided us with the enacted complexity at 11 distinct points in the project (Figure 5).

INSERT FIGURE 3, 4, & 5 ABOUT HERE

Analyzing complexity over time. Given the project narrative and the enacted complexity of each project sprint, we examined how complexity unfolds by exploring the connections between these two complementary ways of seeing the project. The networks of action and indices of enacted complexity for each sprint were mapped onto the temporal narrative. This mapping allowed us to see how patterns of actions corresponded with the temporal unfolding of the project. We then sought to explain these correspondences by iterating between our observations, interviews, and specific actions from the scrum sheets. From this process, we developed a new narrative of how enacted complexity unfolded in ProjectBQ which we present in the next section.

FINDINGS

We report our findings of how action networks enable us to see the complexity of emergent processual phenomena through a narrative of how complexity unfolded in ProjectBQ. Given the "strong process" perspective we adopt, the narrative form enables us to present the richness and detail in our understanding of complexity as a flow (Langley, 2007).

The narrative describes the relationship between enacted complexity, changes to the game design, and delays. The narrative is divided into three parts. The first part describes the relationship between the perceived requirements of the external environment and pressures to complexify patterns of action. The second part describes how the increasingly complex patterns of action led to delays that triggered pressures for simplification. Finally, the third part describes how teams oscillate between simplifying and complexifying.

Overview of Enacted Complexity in ProjectBQ

ProjectBQ was designed as an educational game to teach teenagers how not to succumb to peer pressure in high risk situations. The game was themed as a fantasy game where the hero protagonist is a mouse that is attempting to protect his tribe from the corrupting influence of the villain antagonist. Players progressed in the game by visiting new worlds to battle enemies. Battles were turn-based and were won by whether the player picked the right move that would best counter the one chosen by the computer. Although the game had “fantasy characters”, players had to make decisions based on real world situations. Describes Producer1,

> “It is not direct messages saying, “Don’t do drugs.” What it’s saying is, “Here are some situations that you’re not going to be comfortable with in real life. Here are responses and ways in which you can handle those situations without feeling like a nerd or an outcast, or like you’re going to lose your friends or things like that.” (Producer1)

Ensuring that these messages are salient, yet incorporated into the game with subtlety, while also ensuring that the game was fun was not an easy task. Our findings pointed to ProjectBQ as one that was plagued by frequent design changes, delays, and poor game mechanics. There were also major staffing changes with project leads being replaced and senior developers added to the team in the middle of the project. Three months later, the lead designer’s employment with GameSG was terminated, four months before the end of the project.

The aggregated index of enacted complexity for ProjectBQ across all the Sprints was 18.1 (Figure 3). Over time though, Figure 4 reveals varying patterns of complexity enacted within each sprint. The indices of enacted complexity for ProjectBQ over each three-week sprint cycle are shown in Figure 5. Enacted complexity increased to 4.64 in Sprint 4, declined sharply to 1.56 in Sprint 5, and increased again to 5.80 in Sprint 8. This was followed by a gradual decrease over the remaining sprints (Sprint 9: 3.16, Sprint 10: 2.04, Sprint 11: 1.80).

Part 1: Figuring out Design by Complexifying Patterns of Actions

The patterns of enacted complexity show that in Sprint 1 (Figure 4a), there were not many co-dependencies between the Art and Tech functions, with the exception of Tech Revision and Art Building. The artists are mainly focused on building "Gold Spike" (Sprint 1, Thread 3) which is an early prototype for developers to become familiarized with the production process so that they can anticipate potential pitfalls when they ramp up production. Meanwhile, Design and Tech are both conducting "Research" (Sprint 1, Thread 6). Design is engaged in "Game Research" (Sprint 1, Thread 6, ID 6) which involves understanding the mechanics of similar games such as "Sims Mobile, Princess Maker, My Life as a King, and other pet games" (Sprint 1, Thread 6, ID 6). Tech is researching "server hardware and software needed for the project" (Sprint 1, Thread 6, ID 10) and "networking libraries" (Sprint 1, Thread 6, ID 11). It is common for activities to revolve around experimentation at this stage because it helps designers to understand the game mechanics and make decisions about the features and functionalities that the game will have. Producer1 explained the process at this stage as follows:

> "[The designer] felt that we should have a pre-production phase. Your designer needs a pre-production process because they need to figure out what the game is and then they need to start designing it before the tech people come in and start building it. You can't build something that hasn't been figured out yet." (Producer1)

However, the technical lead, was concerned that the project might have been overscoped and the team would not have enough time to produce the promised deliverable to the client. As a result, the tech lead lobbied the producer to request from senior management for more developers to join the team as early as possible so that they could start building the game quickly. Having a large headcount assigned to the project at the start led to pressures for the team to be productive. Compared to a team with a smaller headcount, the bigger team incurred higher overhead costs earlier in the process which diminished the project budget at a faster rate. Explained the Tech lead, Programmer1,

> "We needed [design] to make some big decisions now about things so this team of content creators who are on the project full time have something that they can build." (Programmer1)

However, the designer's lack of experience with this genre of games meant that he was unfamiliar with the design elements that made for a compelling game. To resolve these ambiguities, he required more time for pre-production work to figure out these issues. As a compromise, the Tech team identified parts of the project that they could work on before the design had been finalized. For example, Tech worked on implementing core functionality on the "Game Engine" (Sprint 2, Thread 8), which was the software that controlled graphics and animations in the game. Meanwhile, Art continued working on the "gold spike" for "Training/Battles" (Sprint 2, Thread 16) which depended on "Storyboards of the Battle" (Sprint 2, Thread 16, ID 7) from Design. According to Producer1, this compromise was made in the following way:

> "What ended up happening was that we had some concepts and ideas of what we were building. Tech was starting to build certain things. Art still did not have an idea of what look and feel we were going for [and] were trying to work that out with design … [and] with tech." (Producer1)

Evidence for this pattern of dependencies was reinforced in the action network for Sprint 2 (see Figure 4b). To give Design more time to "figure out what the game is", Tech focused on building core functionalities of the game engine that were independent of designer's decisions.

This explains why there are no edges between Tech and Design in Sprint 2. However, artists were still co-dependent with designers because the former required Design's "Storyboards" to develop the "look and feel" for the gold spike on "training and battles" (Sprint 2, Thread 16).

Despite their attempts at relieving the pressure on Design to make "big decisions", the tight interdependencies between functions meant that there were limits to how independently Tech and Design could function from each other. At some point, Design needed to make these decisions so as not to cause further delays to Tech's progress. This is evident from the action networks which show the co-dependencies between Design, Art, and Tech in Sprints 3 and 4 (Figure 4c and 4d). Explained the Art lead, Artist1:

> "As we got more work done, we realized that this design wasn't working. This spec needed to change which forced a rewrite of tech. It happened a lot with UI (user interface) and it happened a lot with some of the other core mechanics, like the burrow and combat." (Artist1)

Changes to the design meant that new design elements had to be "figured out", which led to more iterations between Design, Art, and Tech (i.e., more edges) as well as the addition of new tasks (i.e., more nodes) to implement these "rewrites". Consistent with Artist1's comments, the scrum sheets revealed that many of the co-dependencies between the Design, Art, and Tech functions emerged from the "Combat" story. Between Sprints 3 and 4, various parts of the "Combat" story were at different stages of development - some were in the early stages of experimenting, while others were in the later stages of testing and revision. For example, of an experimenting task that Design was assigned to in the Combat story was "Influences for Combat" (Sprint 4, Thread 22, ID 56); while an example of a later stage testing task for Tech was "2nd pass on enemy AI" (Sprint 4, Thread 22, ID 59). The higher number of nodes and edges caused by changes to game design was associated with higher degrees of enacted complexity, which was 2.68 in Sprint 3 and 4.70 in Sprint 4.

Part 2: Responding to Delays by Simplifying Patterns of Actions

Instead of a linear flow of action from Design to Art functions, and from Experimenting to Building then to Testing for task types, the action network in Figure 4c and 4d show that the flow of actions is iterative with co-dependencies between multiple nodes. This iterative process also meant having to discard assets that had already been built. Over time, these changes frustrated developers who became more reluctant to fully commit to completing their assigned tasks. Artist1 describes the artists' reaction to this situation:

> "The guys get to a point where Art wouldn't actually be making any final art for anything because we weren't sure [about] spending that time. Let's say that it's going to take you ten hours to make a final piece of art today. Well guess what? No one's ever going to get more than five hours at any task, because we don't know what's going to get cut. If you have 20 things you need to do, instead of spending ten hours on each of those tasks, we're going to go through all of that for five hours. Hopefully, we'll have something to show for you." (Artist1)

Experiencing frequent changes in design created the expectation that more changes were forthcoming. According to Artist1, this expectation, coupled with having "twenty things you need to do" led to a more cautious approach where tasks would only be partially completed. By being more cautious about committing fully to a task, developers would be less upset by the decision to discard or change a feature. But if the feature was retained, then they just needed to put in an incremental amount of time to complete it. While a cautious approach to production made sense from the individual's perspective, it created uncertainties in the production pipeline and led to delays because it was not clear if they would have “something to show” at the end of the sprint. Thus, a result of design changes was the increased propensity for delays arising from disruptions to the sequential flow of progress. Moreover, developers were demoralized by these changes which led them to perform tasks only to partial completion, exacerbating delays to progress.

The link between enacted complexity and delays can be illustrated from a scrum in Sprint 9 (Nov 10, p. 110), where team members reported being blocked on a number of tasks. The Design Lead is blocked on all the different types of “Powers Scripting” tasks. He is waiting on the Tech team to complete “Heal & Status Effect Support”. Programmer RH has a task of creating a “Rough of Burrow combat background” is blocked by Artist DH who is completing the map. However, DH is blocked on performing critical tasks of this feature as he is awaiting the project leads to make a decision about the “Tunnels” feature. Tech1 is blocked by Tech3, who is waiting on Design to finalize how players will “level up”. However, Design1 the lead designer does not have the capacity to look into this decision now, so the task falls to KK. Tech3 also reports that he is blocked by Design1 on a number of tasks - “Integrate animation system updates” and “Integrate new UI art”.

As these delays accumulated over time, deadline pressure on the team correspondingly increased. In the above incident, the team was two weeks away from the feature lock deadline for the gamma build. Developers responded to these pressures by removing tasks - in effect, reducing enacted complexity by removing nodes and edges in the action network.

As shown in Figure 5, the index for enacted complexity fell from 5.80 in Sprint 8 to 3.16 in Sprint 9. This change can be attributed to intentional efforts taken by developers to reduce complexity. For example, in the Nov 10 scrum in Sprint 9, tasks such as “Cut up the mage mouse" and "Cut up the warrior mouse" were postponed until "post-gamma" (Field notes, Nov 10, p. 110). These tasks were brought back in Sprint 11. During the Dec 2 scrum, Tech1 had to decide which features to "drop" for the upcoming gamma build. He emphasized that the build just needed to be "good enough" (Field notes, Dec 2, p.118).

Comparing the action networks of Sprint 8 (Figure 4e) and Sprint 9 (Figure 4f), one of the main differences is that in Sprint 9, Design Building is less connected to other nodes. This suggests that developers are no longer iterating between design and production. These decisions have been finalized and developers are focused on coordinating actions to complete production of planned features.

Part 3: Oscillating Between Simplifying and Complexifying

While reducing features allowed them to meet their deadline for the gamma build, it also meant having to compromise on the functionality and mechanics of the game. Artist1 explained how these events unfolded in ProjectBQ when the studio head finally had a chance to preview the game:

> "End of March, beginning of April when [the studio head] came in and just basically, professionally, very professionally asked, "What the fuck? What's going on? This isn't working. … He is looking at this halfway polished game that doesn't have much of the functionality that we wanted initially. He gets freaked out." (Artist1)

To compensate for this shortfall in quality, core aspects of the game were redesigned. Explained Artist1,

> "[The studio head] makes this massive push to change the game mechanics - to better the existing game mechanics, to better the user flow, the user experience; which then has a lot of stuff getting either cut or tossed or added." (Artist1)

Not only did this “massive push to change the game mechanics” lead to more tasks being added, it also meant that other features were removed, which demoralized developers further and added to their disgruntlement with the project. Artist1 describes these changes and his reaction in the following quote,

> “He took the concepts and shook it upside down. […] These original ten ideas that we had blew up to 20, then went up to 25. He cut them down to 15, and then they're different from the original ten at this point. It was really shitty.” (Artist1)

This phenomenon where simplifying created more complexity and led to further delays was also evident from changes to the Combat system, as described by Programmer3.

> "You build a combat system to handle up to three mice versus three enemies [3v3], we wanted that early on, and then later we say, “Screw it. We’re just going to 1v1, so we can’t afford to do more.” All that work you put to handle the 3v3 is wasted and worse, you get into this terrible situation where you paint yourself into a corner where you can’t fix the code to properly implement 1v1 because you just don’t have time.” (Programmer3)

The ProjectBQ team thus found themselves oscillating between complexifying and simplifying patterns of action. This frustrated developers and stymied the team’s progress on accomplishing project goals. More drastic efforts to address the problems facing this team were taken that saw the replacement of one of the project leads with another designer, the addition of a senior artist to help the Art lead with the artwork for marketing materials, and another senior producer supporting the team as an Associate Producer. Unfortunately, the problems continued eventually leading to the termination of the lead designer on the project.

DISCUSSION

We draw on our findings to develop a conceptual model of the dynamics of enacted complexity (Figure 5). Based on our observations at GameSG, complexity did not look like a static property or characteristic of an idealized project. Rather, it was a dynamic phenomenon in which patterns of action oscillated between high and low levels of enacted complexity. This dynamic view of complexity highlights the emergence of complexity through endogenous actions (Poulis & Poulis, 2016), which adds to the conventional view of complexity being essentially determined by variation, selection, and retention (VSR) processes that are exogenous to the organizing entity (Ashby, 1956). We elaborate on our conceptual model and discuss these ideas in the remainder of this section.

Project Complexity Varies by Orders of Magnitude Over Time

Our first and most important finding is shown in Figure 5: the complexity of ProjectBQ varies by nearly five orders of magnitude over time. Like the Richter scale for earthquakes, or the Saffir-Simpson scale for hurricanes, the index of task complexity shown in Figure 5 is *exponential*. In sprints 1, 5, and 12, complexity was at its lowest. In sprint 4, and again in sprint 8, the project became thousands of times more complex. Based on observed actions, the participants in those sprints were enacting thousands of time more possible pathways (sequential combinations of actions) than in the other sprints. This variation was definitely problematic for ProjectBQ.

Unlike our measurement scales for natural phenomena like wind, we do not have a valid basis for saying whether the levels of complexity in ProjectBQ would be cause for concern in other settings. This is new terrain for organizational science. Ryan et al (2016) applied the same methodology in a preliminary analysis of routines in dermatology clinics, but did not have sufficient data to examine a connection between complexity and patient outcomes. Hansson et al (2018) are using it to examine the complexity of team decision-making processes. It will take time before we accumulate observations and understand the implications of enacted complexity. At least now we can see how fast the wind is blowing and see that it can change dramatically over time.

Endogenous versus Exogenous Drivers Of complexity

Prevailing conceptions of complexity in management view the configuration of organizational structures as an adaptive response to exogenous, environmental stimuli (Barreto, 2010; Boisot & McKelvey, 2010; Child & Rodrigues, 2011; Eisenhardt & Pienzunka, 2011; Ndofor, Sirmon, & He, 2011; Poulis & Poulis, 2016; Weick, 2007). This adaptive response was revealed in our data from how the leads in ProjectBQ felt pressured to start with a big team of developers to meet ambitious project goals and tight deadlines. But these conditions simply set the context and do not explain how complexity emerges. We developed these insights from our analysis of the events and actions that unfolded.

At the start of the project, many higher-level design features had not been made and there was still a need for designers to figure out whether an idea was feasible and fun. This step involved building prototypes ("gold spikes") to objectify the desired gaming experience and better allow the team to explore the dependencies between functions in the production process. However, the pressure for developers to start work quickly led them to work on tasks based on unvalidated assumptions about the design. As design decisions became clearer through prototyping, some of the work that had already begun had to be revised or discarded due to changes in task specifications. This pattern manifested in the action network as the addition of new actions (nodes) and dependencies between actions (edges) which led to an increasing index of enacted complexity over sprints.

While our narrative of ProjectBQ is consistent with the prevailing conceptualization of complexity as a response to the external environment, we show how this emergence occurs endogenously through emergent actions, rather than from VSR processes exogenous to the project team. Explanations based on VSR refer to population level phenomena (Van de Ven & Poole, 1995); if a process variation occurs in some set of projects, and those projects are more successful, that variation is more likely to be replicated in subsequent projects. While VSR processes may be occurring at GameSG, we observed complexity varying in orders of magnitude within a single project as the project was being carried out. External forces cannot explain the fluctuations in complexity that were evident in our data.

Conceptualizing the Dynamics of Complexity

Our analysis of ProjectBQ provides the basis for a conceptual model for these complexity dynamics (Figure 6). In ProjectBQ, an increasing level of enacted complexity did not necessarily translate into progress on accomplishing goals even though more work was being done. The interdependencies between functional roles increased difficulties in coordinating activities (Okhuysen & Bechky, 2009), which delayed progress and adversely affected output quality (Figure 6, arrow 1). Addressing these shortfalls in quality meant having more iterations between different functions and task types, which increased complexity and further delayed progress.

As the realization of falling behind schedule increased, so too did the pressure to reduce complexity (Figure 6, arrow 2). In order for ProjectBQ developers to meet intermediate deadlines, decisions were made to simplify game design by reducing project scope, features, and functionalities. These decisions allowed task requirements to be more clearly defined, reducing the need to iterate between functions. Developers were more focused on completing tasks within their respective functions. With a more linear workflow and less iterations between functions and task types, the action network contained fewer edges between nodes which manifested as declining levels of enacted complexity. This decline was due to accumulated pressures for reducing complexity (in the form of deadline pressure) reaching a tipping point, rather than because of environmental fit.

The story, however, does not stop here. ProjectBQ experiences another surge in enacted complexity. Our data revealed that prior attempts to simplify the workflow had repercussions on game design (Figure 6, arrow 3). For example, if the “microtransactions” feature is cut from the game, the designer would need to make fundamental changes to the game, such as the reward system, how players “level up”, and revising the story so that it is still compelling, coherent, and consistent with other parts of the game.

These design changes led to new features added and assets that had already been built to be revised. Integrating these new additions with existing assets required developers to coordinate across functions which led to an increase in nodes and edges in the action network (Figure 5, arrow 4). Thus, complexity increased because of the accumulated pressures for enhancing complexity in the form of redesigns that stemmed from efforts to simplify deliverables.

These variations in complexity over time cannot be accounted for by prevailing explanations of complexity that rely on VSR processes. Our conceptual model of the dynamics of enacted complexity (Figure 6) posits pressures for reducing and enhancing complexity as mechanisms for these changes. Over the duration of ProjectBQ, the influence of deadlines and redesigns wax and wane, resulting in the observed oscillations in level of complexity.

INSERT FIGURE 6 ABOUT HERE

Implications for Theory Development

We next consider the theoretical implications of shifting our view of complexity from a static entity, to a dynamic, emergent process enacted by agentic actors. First, this perspective allows scholars to explicitly consider dynamism in theories of complexity. It sets the stage for further theorizing of the antecedents and consequences of temporal trajectories of complexity. In this research, we showed how pressures for complexity reduction and increase tilted the balance towards adding and removing nodes and edges. What other factors might enhance or mitigate these pressures? Understanding these factors will enrich our explanations for differences in the temporal trajectories of complexity.

In addition to examining the antecedents of complexity, we can also theorize about how the pattern of complexity emergence affects important organizational outcomes. As evident in ProjectBQ, wide swings in the magnitude of complexity were detrimental to the timeliness and quality of project deliverables. On the other hand, varying patterns of complexity could reflect an agility in responding to dynamic circumstances or environmental perturbations (Brown & Eisenhardt, 1997; Van de Ven, Polley, & Garud, 1999).

A second implication for theorizing is that tracing and quantifying processual phenomena at the level of actions enables researchers to identify how these phenomena emerge and flow. We can visualize, measure, and quantify differences in characteristics of the flow. Complexity is just one such characteristic. To the extent that organizational becoming (Feldman, 2000; Tsoukas & Chia, 2002; Weick & Quinn, 1999) manifests in the organization's temporal trajectory (Hernes, 2017), being able to see this trajectory is a step towards theorizing about differences in these trajectories. Such theoretical developments are an initial step towards bridging the divide between variance and process theories (Mohr, 1982; Van de Ven & Poole, 1995) because it allows for the consideration of questions about the processual flow, such as how variations in processes (e.g., pattern of complexity) explain important organizational outcomes.

Limitations

Even as we move from treating complexity as a static phenomenon, our attempt at seeing the "flow" of complexity is still a compilation of static snapshots. The flow we see is not completely smoothed. It is more like a story board or a slide show than a finished movie. It is likely that changing the timeframe or the "exposure time" of the picture could generate a different sense of flow. While future research could examine the question of temporal granularity, our intent in this paper is to highlight the potential of moving from a static to a dynamic view of complexity.

In recreating the task sequences in our data, we had to assume that tasks were performed sequentially. However, some tasks were performed concurrently but we were not able to capture such relationships with current methods. To the extent that concurrent activities are interdependent, this method is likely to understate complexity. Nevertheless, the general trajectory of complexity was consistent with the project narrative, which was based on other data sources. Both of these limitations -- temporal granularity and concurrency -- point to the importance of having multiple sources of data to contextualize and interpret processual phenomena, as we have done here.

CONCLUSION

Our research demonstrates how bringing actors and the context into the picture allows us to see complexity as an agentic enactment influenced by the interplay between interdependent actors, traits, and structures (Giddens, 1984; Hærem et al., 2015; Poulis & Poulis, 2016). In this view, complexity is an inherently dynamic phenomenon that emerges through the patterning of actions performed by multiple, interdependent agentic actors. Actions are connected to one another in time, and it is this patterning of actions that constitutes complexity. Complexity is thus best indexed by observable patterns of behavior performed by multiple agentic actors. Showing the patterning of these actions as a narrative network is akin to being able to see how fast the wind blows and not just knowing what its velocity is. Our analyses of these patterns of behavior makes a theoretical contribution by developing a dynamic model of enacted complexity. Seeing complexity as a dynamic, emergent phenomenon generates new questions about the dynamics of complexity, their antecedents, and consequences on organizational outcomes.

REFERENCES

Abbott, A. 2016. ***Processual sociology***: University of Chicago Press.

Alvesson, M., & Sandberg, J. 2011. Generating research questions through problematization. ***Academy of management review***, 36(2): 247-271.

Ashby, W. R. 1956. ***An introduction to cybernetics***. London: Chapman & Hall Ltd.

Banker, R. D., Davis, G. B., & Slaughter, S. A. 1998. Software development practices, software complexity, and software maintenance performance: A field study. ***Management science***, 44(4): 433-450.

Barreto, I. 2010. Dynamic capabilities: A review of past research and an agenda for the future. ***Journal of management***, 36(1): 256-280.

Blau, P. M., & Scott, W. R. 1962. ***Formal organizations: A comparative approach***: Stanford University Press.

Boisot, M., & McKelvey, B. 2010. Integrating modernist and postmodernist perspectives on organizations: A complexity science bridge. ***Academy of Management Review***, 35(3): 415-433.

Brooks, F. P. 1974. Mythical Man-Month. ***Datamation***, 20(12): 44-52.

Brown, S. L., & Eisenhardt, K. M. 1997. The art of continuous change: Linking complexity theory and time-paced evolution in relentlessly shifting organizations. ***Administrative science quarterly***: 1-34.

Campbell, D. J. 1988. Task complexity: A review and analysis. ***Academy of management review***, 13(1): 40-52.

Child, J., & Rodrigues, S. 2011. How Organizations Engage with External Complexity: A Political Action Perspective. ***Organization Studies***, 32(6): 803-824.

Cohendet, P. S., & Simon, L. O. 2016. Always playable: Recombining routines for creative efficiency at Ubisoft Montreal's video game studio. ***Organization Science***, 27(3): 614-632.

Eisenhardt, K., & Pienzunka, H. 2011. Complexity theory and corporate strategy. In P. Allen, S. Maguire, & B. McKelvey (Eds.), ***The SAGE handbook of complexity and management***: 506-523. London: Sage.

Emirbayer, M., & Mische, A. 1998. What is agency? 1. ***American journal of sociology***, 103(4): 962-1023.

Feldman, M. S. 2000. Organizational routines as a source of continuous change. ***Organization science***, 11(6): 611-629.

Feldman, M. S. 2016. Making process visible: Alternatives to boxes and arrows. In A. Langley, & H. Tsoukas (Eds.), ***The SAGE Handbook of Process Organization Studies***: 625-635. Beverly Hills, CA: Sage.

Giddens, A. 1984. ***The constitution of society: Outline of the theory of structuration***: Univ of California Press.

Glaser, B. G., & Strauss, A. L. 1967. ***The Discovery of Grounded Theory: Strategies for Qualitative Research***. Chicago, IL: Aldine de Gruyter.

Hackman, J. R. 1969. Toward understanding the role of tasks in behavioral research. ***Acta psychologica***, 31: 97-128.

Hærem, T., Pentland, B. T., & Miller, K. D. 2015. Task complexity: Extending a core concept. ***Academy of Management Review***, 40(3): 446-460.

Hansson, M., Hærem, T., & Pentland, B. T. 2018. Identifying and describing characteristics of organizational routines as repertoires of action patterns, degree of routinization, and enacted complexity, ***AOM Specialized Conference on Big Data and Managing in a Digital Economy***. Surrey, UK.

Hernes, T. 2014. ***A process theory of organization***. Oxford: Oxford University Press.

Hernes, T. 2017. Process as the becoming of temporal trajectory. ***The SAGE handbook of Process organization studies. London: Sage Publications***: 601-606.

Jarzabkowski, P. 2004. Strategy as practice: recursiveness, adaptation, and practices-in-use. ***Organization studies***, 25(4): 529-560.

Langley, A. 1999. Strategies for theorizing from process data. ***Academy of Management Review***, 24(4): 691-710.

Langley, A. 2007. Process thinking in strategic organization. ***Strategic organization***, 5(3): 271-282.

Langley, A., Smallman, C., Tsoukas, H., & Van de Ven, A. H. 2013. Process studies of change in organization and management: Unveiling temporality, activity, and flow. ***Academy of Management Journal***, 56(1): 1-13.

Langley, A., & Tsoukas, H. 2017. Introduction: Process thinking, process theorizing and process researching. In A. Langley, & H. Tsoukas (Eds.), ***The SAGE Handbook of Process Organization Studies***. Thousand Oaks, CA: Sage.

Mesle, C., & Dibben, M. R. 2016. Whitehead's Process Relational Philosophy. ***The SAGE Handbook of Process Organization Studies***: 29.

Moe, N. B., Dingsøyr, T., & Dybå, T. 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. ***Information and Software Technology***, 52(5): 480-491.

Mohr, L. B. 1982. ***Explaining organizational behavior***. San Francisco, CA: Jossey-Bass.

Moldoveanu, M. C., & Bauer, R. M. 2004. On the relationship between organizational complexity and organizational structuration. ***Organization Science***, 15(1): 98-118.

Ndofor, H. A., Sirmon, D. G., & He, X. 2011. Firm resources, competitive actions and performance: investigating a mediated model with evidence from the in-vitro diagnostics industry. ***Strategic Management Journal***, 32(6): 640-657.

Obstfeld, D. 2012. Creative projects: A less routine approach toward getting new things done. ***Organization Science***, 23(6): 1571-1592.

Okhuysen, G. A., & Bechky, B. A. 2009. Coordination in organizations: an integrative perspective. ***Academy of Management Annals***, 3(1): 463-502.

Pentland, B. T., & Feldman, M. S. 2007. Narrative networks: Patterns of technology and organization. ***Organization science***, 18(5): 781-795.

Pentland, B. T., & Liu, P. 2017. Network Models of Organizational Routines: Tracing Associations Between Actions. In S. Jain, & R. Mir (Eds.), ***Routledge Companion to Qualitative Research in Organization Studies***.

Pentland, B. T., Recker, J., & Wyner, G. 2015. A thermometer for interdependence: Exploring patterns of interdependence using networks of affordances, ***Thirty Sixth International Conference on Informaion Systems***. Fort Worth, TX, USA.

Pentland, B. T., Recker, J., & Wyner, G. 2016. Conceptualizing and measuring interdependence between organizaitonal routines, ***Thirty Seventh International Conference on Information Systems***. Dublin, Ireland.

Pentland, B. T., Recker, J., & Wyner, G. 2017. Rediscovering Handoffs. ***Academy of Management Discoveries***, 3(3): 284-301.

Poulis, K., & Poulis, E. 2016. Problematizing Fit and Survival: Transforming the Law of Requisite Variety Through Complexity Misalignment. ***Academy of Management Review***, 41(3): 503-527.

Ryan, J., Pinto, D., Pentland, A., & Pentland, B. 2016. Complexity thermometer: Unraveling complexity of clinic process. ***Journal of Investigative Dermatology***, 136(5): S26.

Sutherland, J., & Sutherland, J. J. 2014. ***Scrum: The Art of Doing Twice the Work in Half the Time***. New York: Crown Business.

Tsoukas, H., & Chia, R. 2002. On organizational becoming: Rethinking organizational change. ***Organization science***, 13(5): 567-582.

Van de Ven, A. H., Polley, D., & Garud, R. 1999. ***The Innovation Journey***. New York, NY: Oxford University Press.

Van de Ven, A. H., & Poole, M. S. 1995. Explaining development and change in organizations. ***Academy of management review***, 20(3): 510-540.

Weick, K. E. 1979. Cognitive processes in organizations. ***Research in organizational behavior***, 1(1): 41-74.

Weick, K. E. 2007. The generative properties of richness. ***The Academy of Management Journal***, 50(1): 14-19.

Weick, K. E., & Quinn, R. E. 1999. Organizational change and development. ***Annual review of psychology***, 50(1): 361-386.

Wood, R. E. 1986. Task complexity: Definition of the construct. ***Organizational behavior and human decision processes***, 37(1): 60-82.

Figure 1. Frequency of task types across sprints

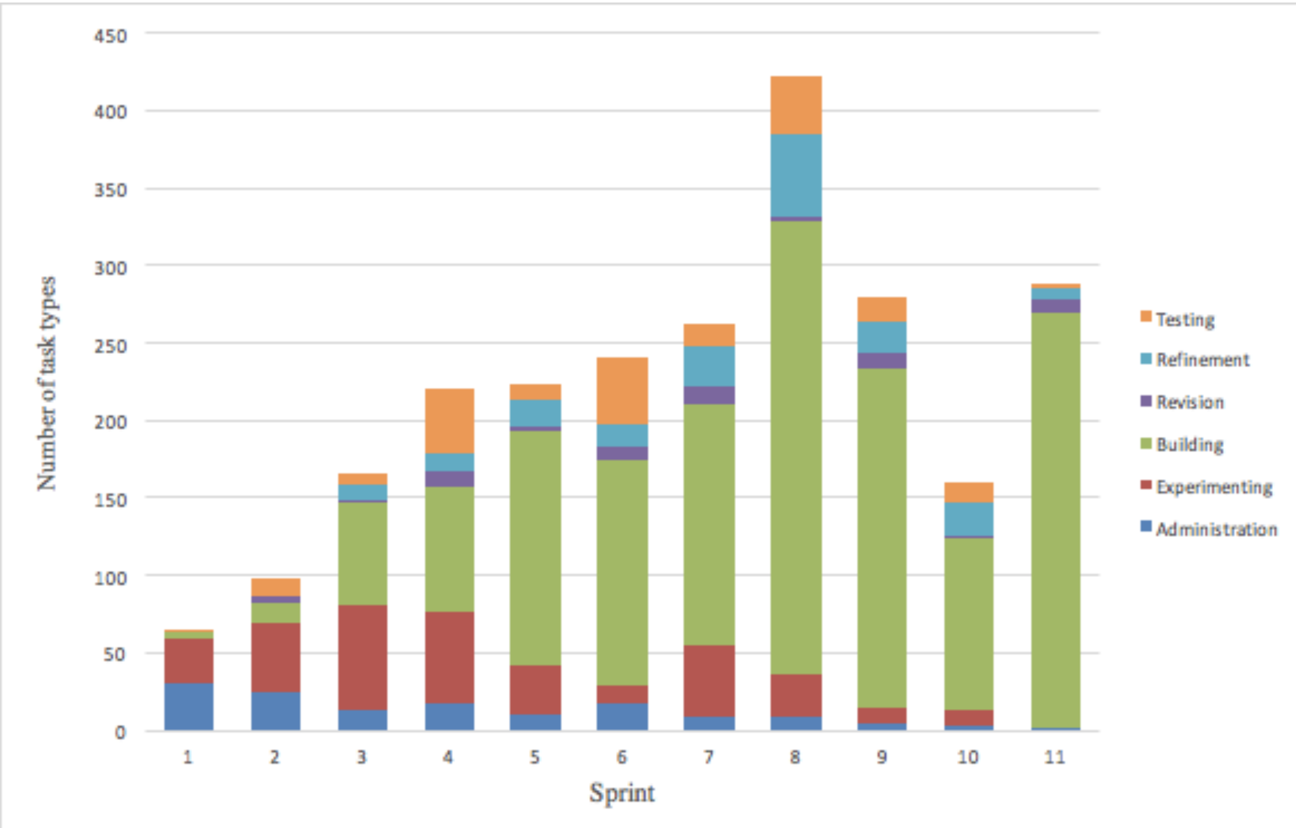

Figure 2. Frequency of roles performing each task across sprints.

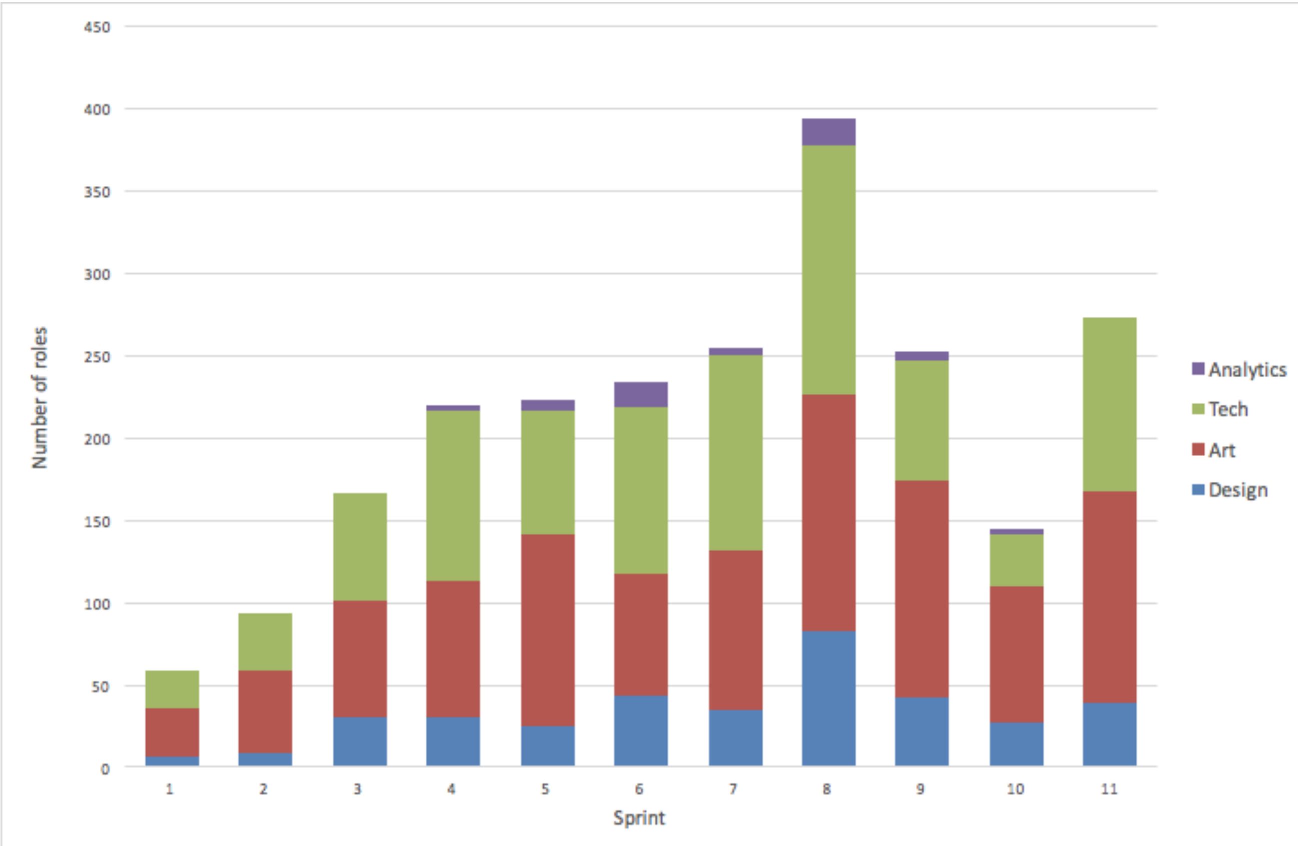

Figure 3. ProjectBQ Action Network.

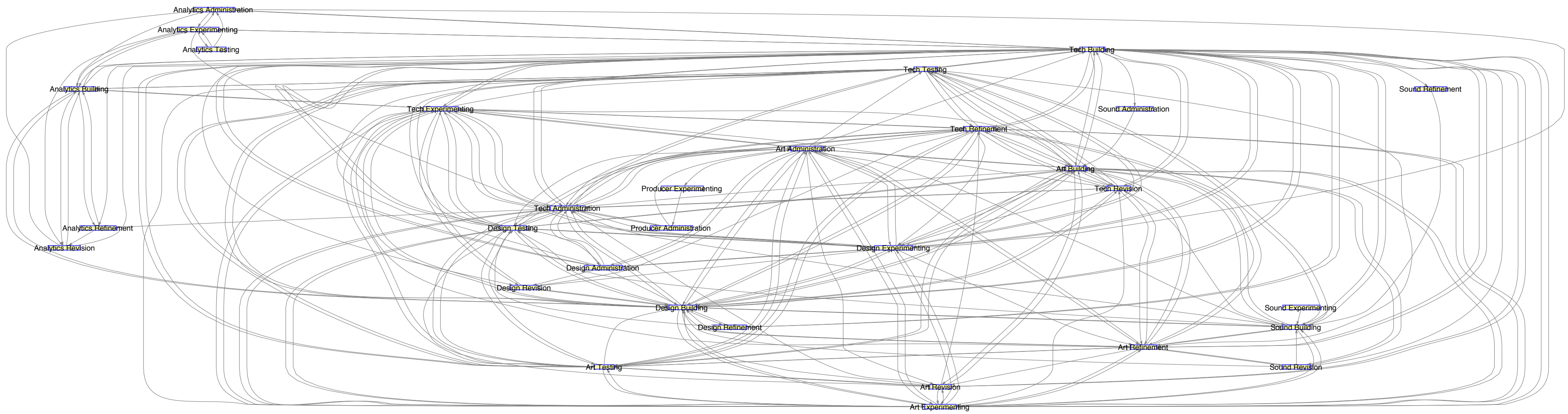

Figure 4a. ProjectBQ Sprint 1 Action Network

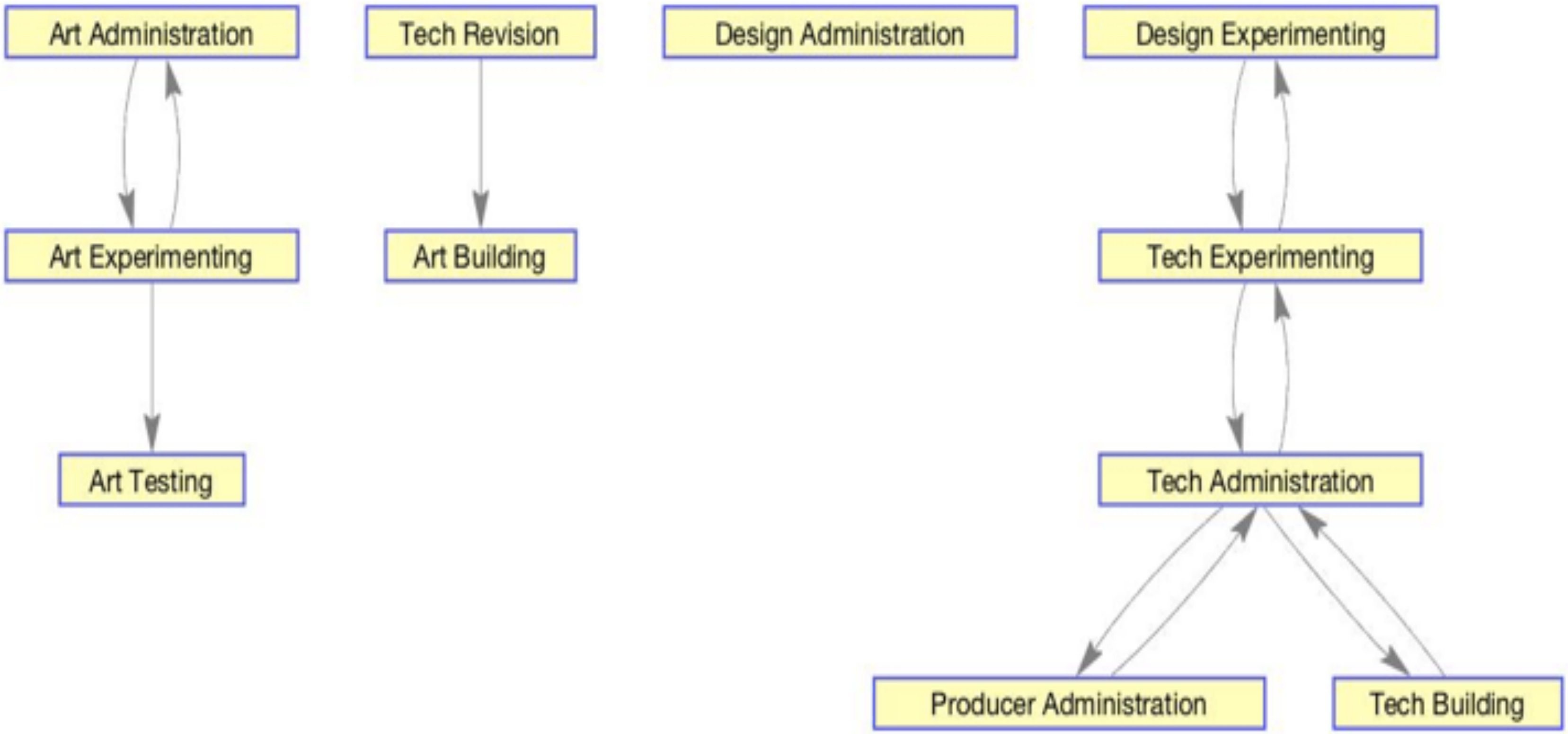

Figure 4b. ProjectBQ Sprint 2 Action Network

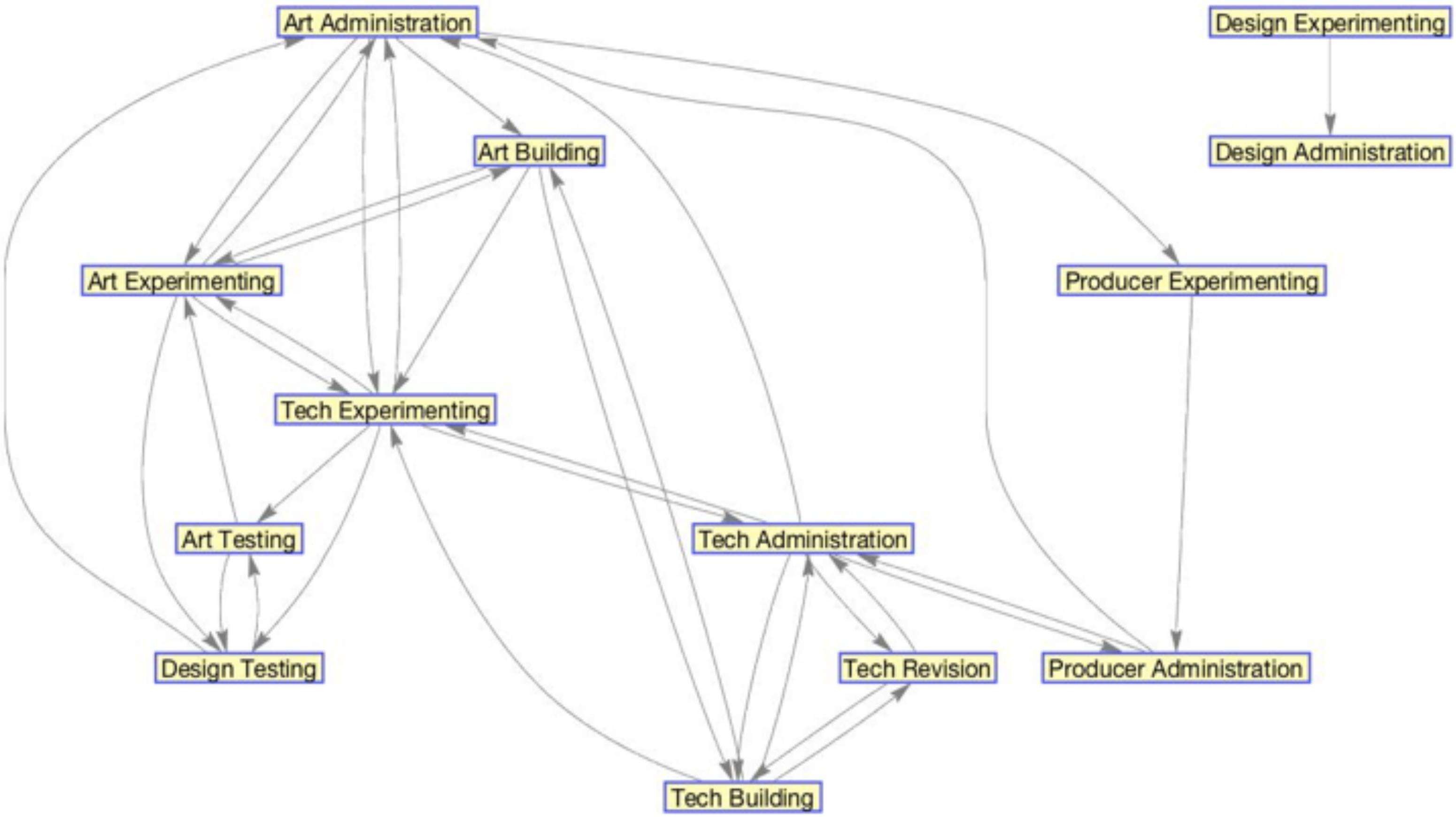

Figure 4c. ProjectBQ Spint 3 Action Network

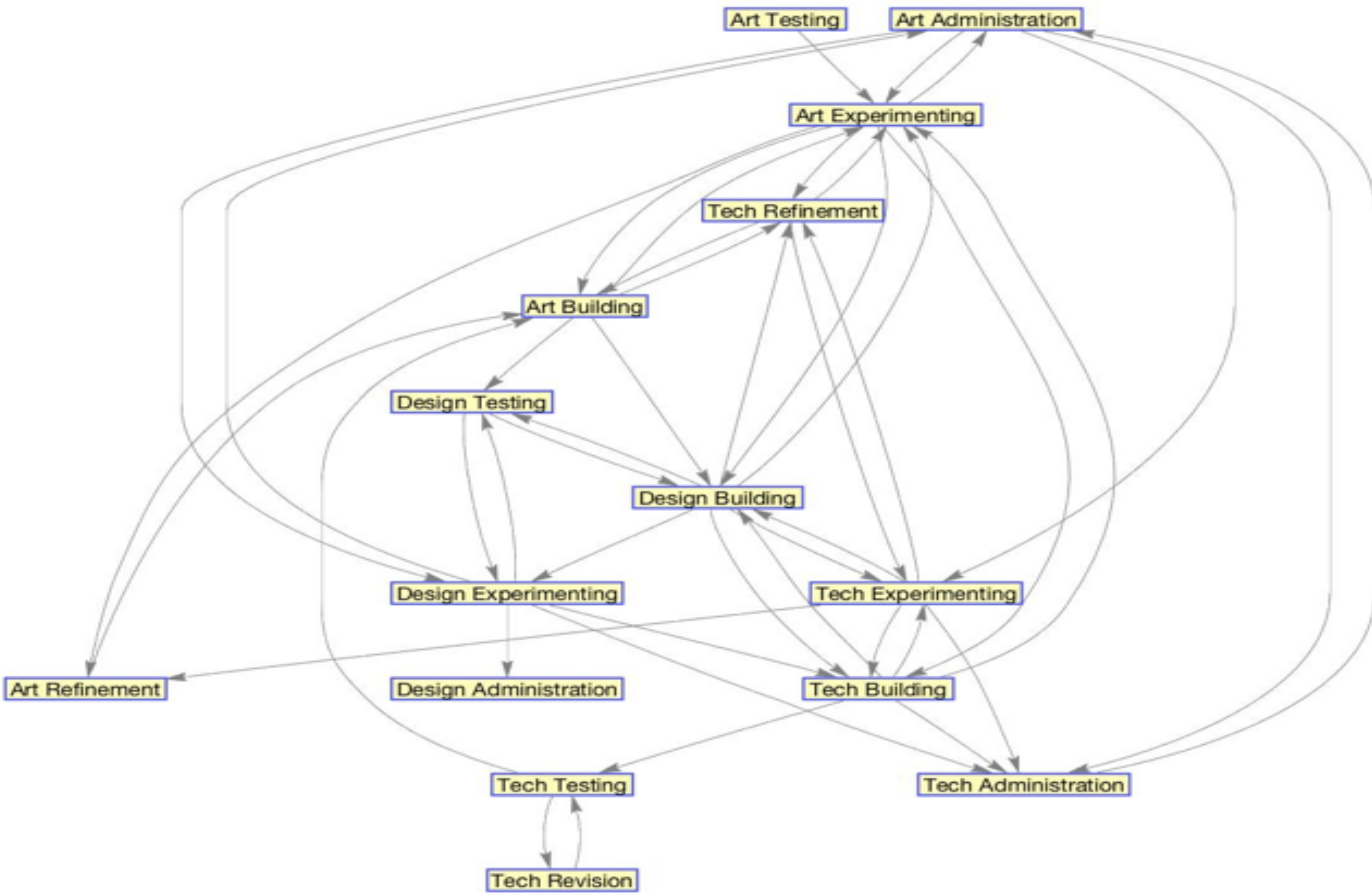

Figure 4d. ProjectBQ Sprint 4 Action Network

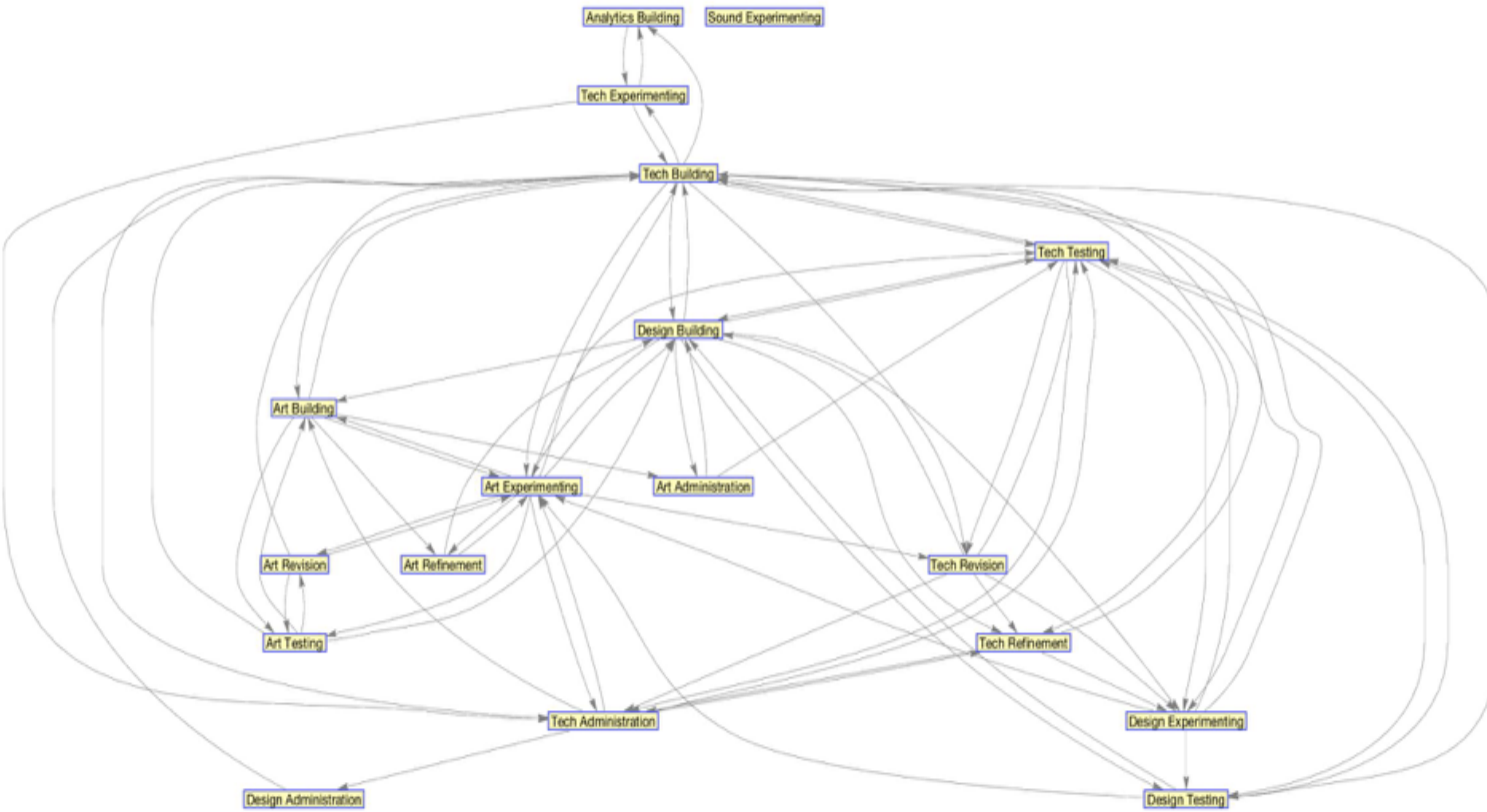

Figure 4e. ProjectBQ Sprint 8 Action Network

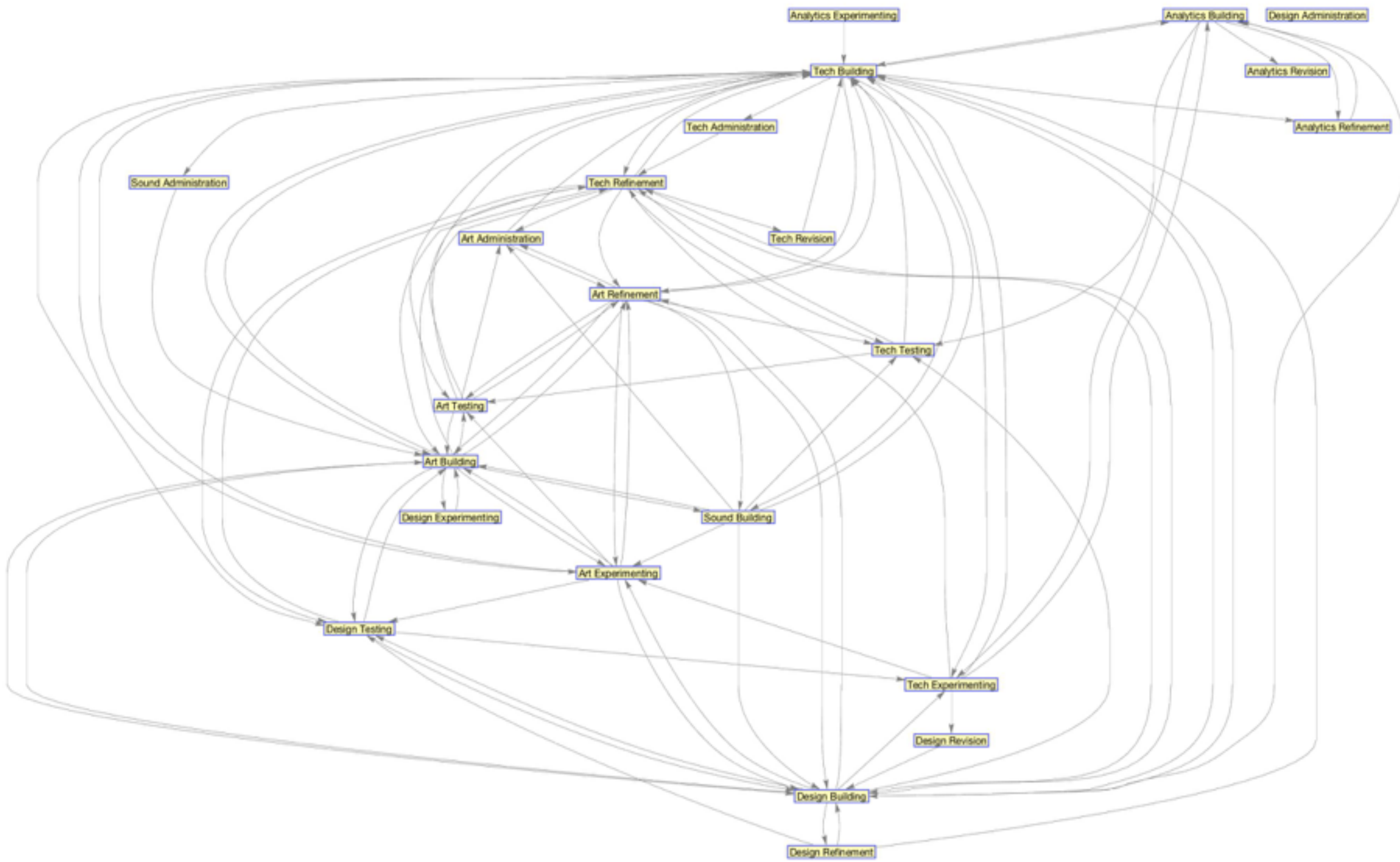

Figure 4f. ProjectBQ Sprint 9 Action Network

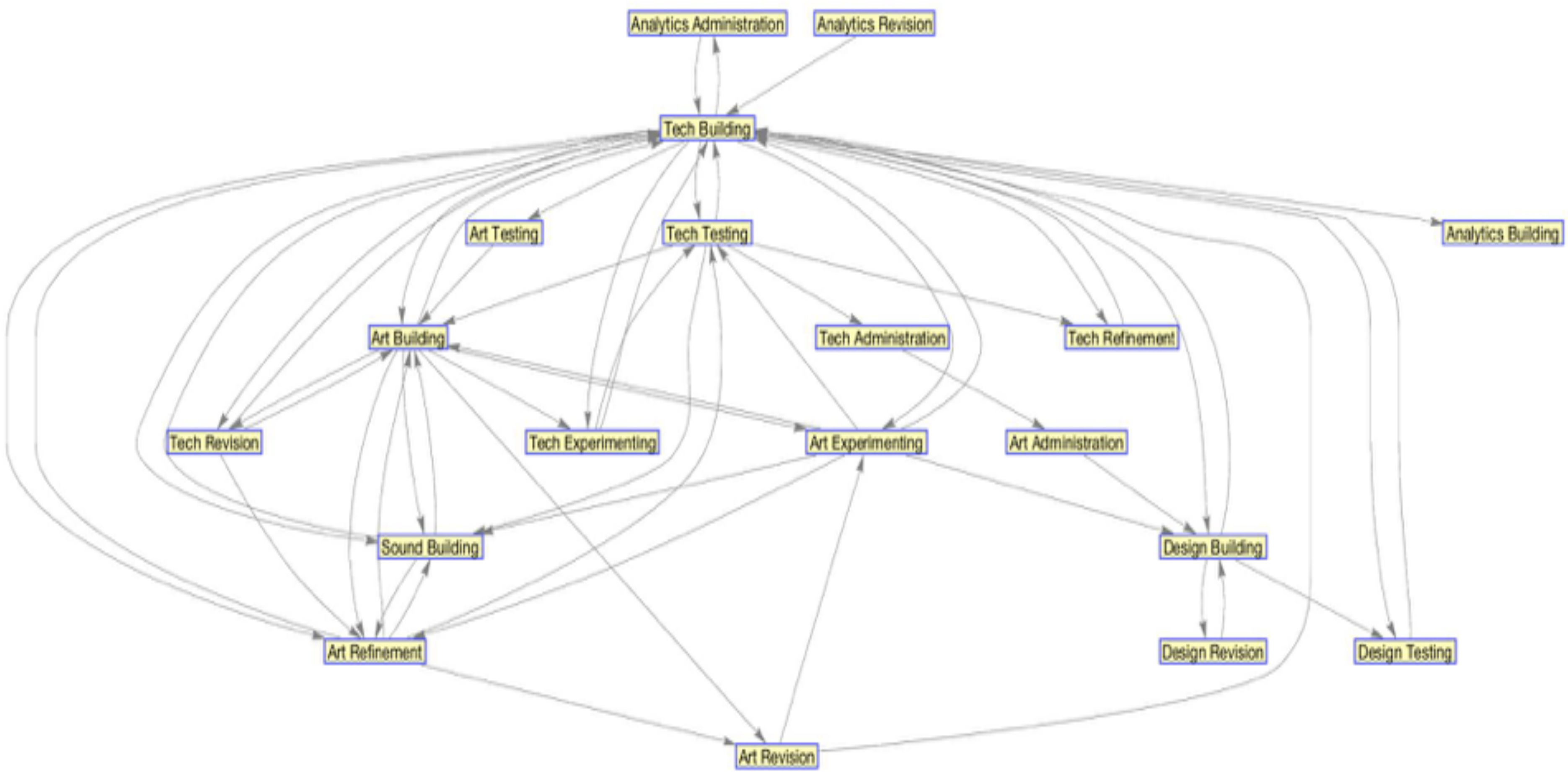

Figure 5. ProjectBQ Enacted Complexity Index for Sprints 1 to 11.

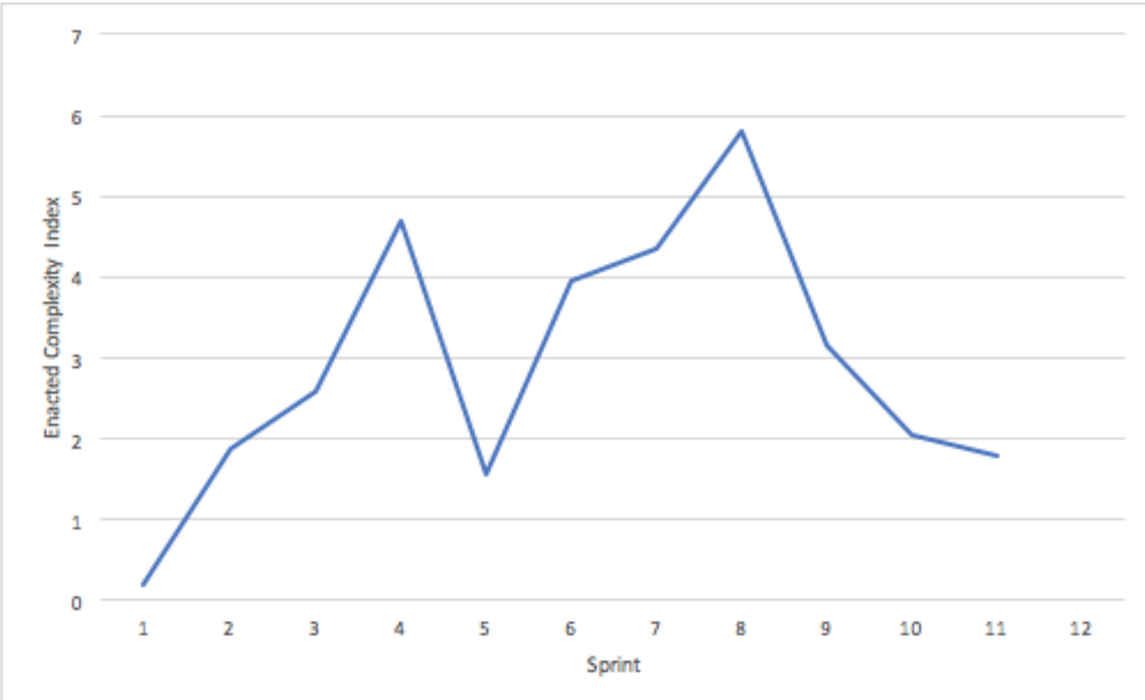

Figure 6. Ongoing work processes

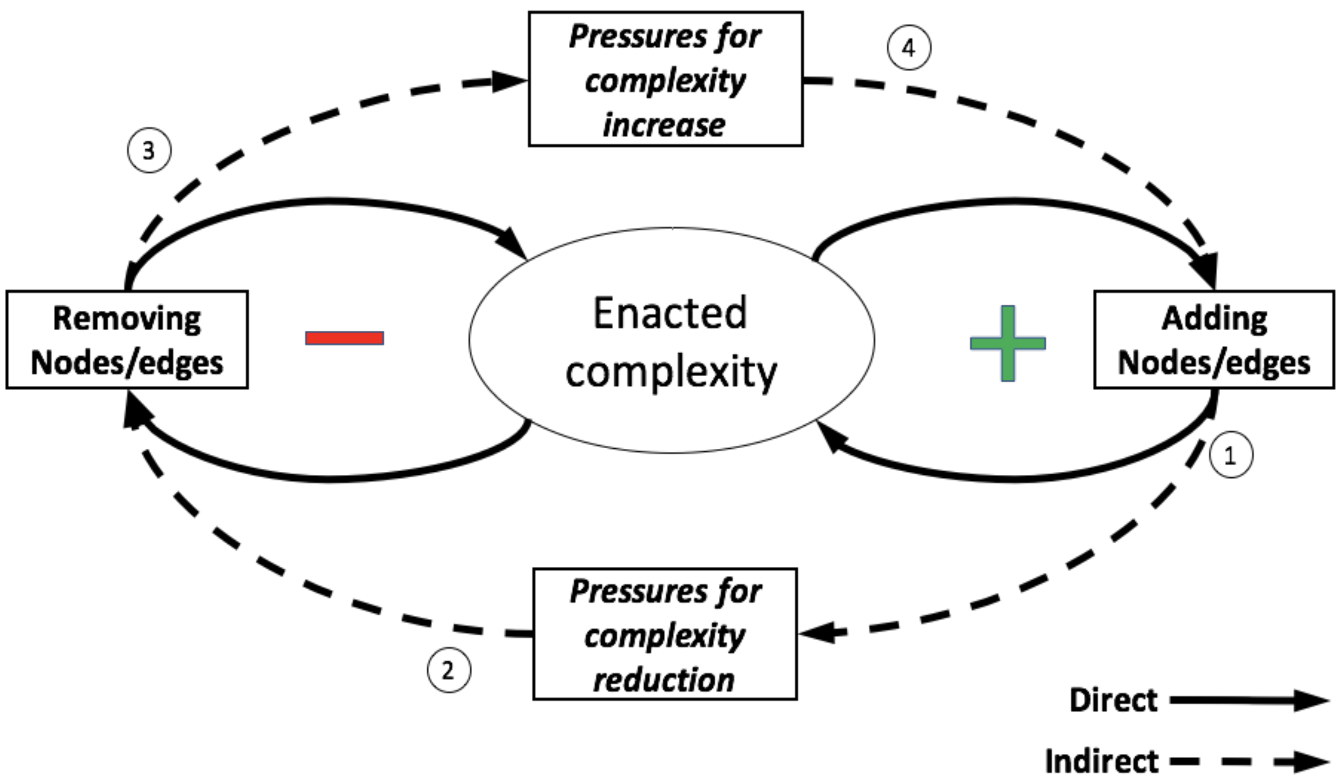

Table 1. Definition of task categories

Category: Final	Category: Round 1	Definition	Examples	Notes
Administration	Administration	Activities that involve planning, organization, coordination, communication with internal or external parties.	Art: Renaming asset names. Tech: Archiving code, setting up	Does not require specialized knowledge (art, software, design)
Experimenting	Experimenting	Activities associated with learning, discovery, building experience or knowledge, addressing unanswered questions.	Art: "What is the mood and feel of..?" Tech: Ramp-up, research Design: Playing games, building experience	
	Conceptualization	Activities associated with defining the form of team output. Includes definition of inter-relationships between components of team output, how output fits with client's other activities (e.g., marketing).	Art: Mockups, concept art, Design: How are rewards given?", "How are rewards spent?", "Narrative that marries story to gameplay"	Manifests as transitional output or boundary objects.
Building	Building	Activities directly associated with producing assets.	Art: Includes modelling and texturing Tech: Writing code from scratch Design: Actual writing up or production of documents	
	Integration	Activities associated with combining different parts of the team output (eg., art assets).	Art: Integrate into engine. Level assets can be integrated by artist, but not characters or props Tech: Asset integration. Export	

Revision	Revision	Activities associated with rebuilding, reimplementation, redesigning or rewriting. Adjustments made to core aspects of output (e.g., code, model, animation) in terms of the relationship between parts.	Art: Changing the theme, not just changing colors/texture. Design: Remove asset/feature. Rewriting code. Improving performance. Bug fixing. Tech: In Mummy, changing economy system due to introduction of RC and VC.	If the relationship between A and B could be specified in an equation, this will involve changes to variables in the relationship, rather than the absolute value of the variables.
Refinement	Refinement	Activities associated with adjusting parameter values of output.	Art: Changing colors, texture Tech: Balancing, code clean up Design: Jeeps need to move slower - they are too hard to hit atm plus the planes need to be moving faster than them for grenovision	
	Fix	Activities associated with rectifying errors.	Art: Redraw or rebuild Tech:Bug fixes. Design: There is a 2k target that sometimes appears on the left side - dunno where this target is supposed to be but appears to be in the wrong spot atm (maybe this is one of the target intended for inside the secret base?) - ID 810 plugin	Closely related to "Tweak", but difference here is that the adjustment is made to some part of output that is broken, or not working as it should. Words like "correct", "error". Result/outcome is unintended.
Testing	Review	Reviewing work before release	First pass, etc.	
	Testing	Activities directly associated with enacting playtests.		Different from QA tests which checks technical integrity of output
	Feedback	Activities related to obtaining or aggregating feedback from playtests or metrics, by clients or users	Tech: Demo, prototype	Related to the event of obtaining feedback
	QA	Activities that involve testing for bugs, errors or edge cases.		Different from playtests