Helping Data Science Students Develop Task Modularity

Jeffrey S. Saltz
Syracuse University
jsaltz@syr.edu

Robert Heckman
Syracuse University
rheckman@syr.edu

Kevin Crowston
Syracuse University
crowston@syr.edu

Sangseok You
HEC Paris
you@hec.fr

Yatish Hedge
Syracuse University
yhedge@syr.edu

Abstract

This paper explores the skills needed to be a data scientist. Specifically, we report on a mixed method study of a project-based data science class, where we evaluated student effectiveness with respect to dividing a project into appropriately sized modular tasks, which we termed task modularity. Our results suggest that while data science students can appreciate the value of task modularity, they struggle to achieve effective task modularity. As a first step, based our study, we identified six task decomposition best practices. However, these best practices do not fully address this gap of how to enable data science students to effectively use task modularity. We note that while computer science/information system programs typically teach modularity (e.g., the decomposition process and abstraction), and there remains a need identify a corresponding model to that used for computer science / information system students, to teach modularity to data science students.

1. Introduction

Data Science is an emerging discipline that combines expertise across a range of domains, including software development, data management and statistics. Data science projects typically have a goal to identify correlations and causal relationships, classify and predict events, identify patterns and anomalies, and infer probabilities, interest and sentiment [1]. Big Data is a related field, often thought of as a subset of data science, in that data science applies to large and small data sets and covers the end-to-end process of collecting, analyzing and communicating the results of the analysis. With the increasing ability to collect, store and analyze an ever-growing diversity of data that is being generated with increasing frequency, the field of data science is growing rapidly.

As a new field, much has been written about the use of data science and algorithms that can generate useful results. Unfortunately, less has been written about the project skills students need to learn in order to be skilled data scientists [2], especially on how these emerging data scientists can work together on a data science project.

One aspect of enabling a team to work well together is by having the team be able to break the project into modular components [3]. A modular approach enables the team to proceed more quickly and effectively [4]. Furthermore, it has also been noted that modularity brings increased flexibility, a better ability to deal with complexity and the accommodation of uncertainty [5]. More generally, it has been shown that leveraging modularity delivers significant benefits within many contexts, such as manufacturing [6] and, perhaps most commonly, software development [7].

The modularity of a solution can be considered as a continuum describing the degree to which the components of a solution can be separated, worked on independently, and recombined [8]. With respect to data science, the use of R [9] is an example of one aspect of leveraging modularity. The Comprehensive R Archive Network (CRAN) contains thousands of "packages", which can be installed and loaded as needed. These packages enable a team to easily leverage modules developed by others, such as using an advanced machine learning module via a function call, and is a key aspect of modularity (and the growth of R).

However, another aspect of modularity, *task modularity*, is concerned with how a data science team breaks down its activities into modular "chunks of work" that can be worked on in parallel, in a coordinated manner. One important benefit of task modularity is that it helps reduce the need to coordinate details of a team member's work with other team members.

Since data scientists need to work on complex problems, providing a framework for data science students to effectively use task modularity should be a key aspect of data science education. Unfortunately, there are few studies exploring team process effectiveness within a data science context. There has also been minimal research on how to best develop modular thinking in the students who will become the next generation of data science practitioners.

However, there has been a recent study that demonstrated that the Kanban process methodology was a promising approach for collaboration and coordination in data science teams [10]. Unfortunately, that study, or any other study, did not explore how to help data science teams increase their task modularity. With this in mind, we choose to explore modularity within a Kanban process methodology context.

Specifically, our research explores if using a Kanban data science project methodology improves a student's ability to use modular concepts, as compared to the baseline situation that is how most data science student teams currently work, that is, without a well defined process methodology. Thus, we focused on the following research questions:

RQ1: Do data science students naturally apply task modularity while working on a data science project?

RQ2: Does using a Kanban process methodology improve modular thinking in data science students and lead to improved task modularity?

To address our research questions, we report on a mixed method study that explores students using the Kanban process methodology and evaluates if the methodology impacts a student's ability to think in a more modular manner.

The rest of the paper is structured as follows. In Section 2, we review modularity in project management, software engineering, as well as in a data science context. Section 3 describes the methodology for our study. Section 4 discusses the findings from our study. Finally, section 5 presents a synthesis of our research results and section 6 discusses limitations and possible next steps.

2. Background

To provide context for our exploration of task modularity and the Kanban process methodology, this section provides background on the need for improved data science team coordination, project modularity, the growing use of Kanban in the classroom, as well as the current research that focuses on data science education.

As previously noted, one key enabler of team coordination is to be able to decompose a project into modular tasks. Because there has been minimal research reported on data science team collaboration or a data scientist's use of task modularity, in this section we also explore related domains. However, it must be noted that while data science projects do have parallels to other domains, there are also differences as compared to these other types of projects. For example, Chen, Kazman & Kaziyev [18] argue that the use of agile techniques for data science is new and necessitates careful adaptation, as it is dramatically different from smaller, more traditional, data analytics efforts. Furthermore, compared to software development, data science projects have an increased focus on data, what data is needed and the availability, quality and timeliness of the data [1, 19, 20]. Thus, while there are some parallels to other domains, one cannot assume findings in those other domains will be applicable within a data science context.

2.1. The Need for Improved Coordination

Vanauer, Bohle and Hellingrath [11] noted the lack of an empirically grounded big data science methodology. Hence, not surprisingly, it has been observed that most data projects are managed in an ad hoc fashion, that is, at a low level of process maturity [12]. Indeed, it has been argued that data science projects need to focus on people, process and technology [13,14] and not just on algorithms used by data scientists.

Thus, the need for more guidance is recognized with respect to how data scientists can best work together; for example, a Gartner Consulting report advocates for more careful management of the analysis processes, though a specific methodology is not identified [15]. Chen, Kazman & Matthes [16] studied 23 large enterprises and confirmed this gap. This gap is re-enforced by Espinosa and Armour [17], who noted that the main challenge in a data science project is task coordination.

2.2 Task Modularity

We note that the benefit of task modularity is that it supports complex problem-solving by enabling a team member to focus on smaller challenges, rather than needing to focus on the entire problem [21, 22]. When using a modular approach, one leverages a general set of design principles that involves breaking up a problem into discrete chunks [23] and "building a complex product or process from smaller subsystems that can be designed and worked on independently yet function together as a whole" [24].

Hence, task modularity is likely to be import to data science students due to the benefits of decomposing tasks and allowing different team members to work on different aspects of the project, similar to what is done in software development projects. In other words, enabling or improving

modularity can provide data science students with a mechanism to improve team effectiveness.

To better understand the potential importance of task modularity within a data science project, we first explore modularity in the fields of project management and software development.

2.2.1. Modularity and Project Management. Task decomposition is specifically addressed in the Project Management Body of Knowledge, sometimes known as PMBOK [25]. The Project Management Body of Knowledge defines a work breakdown structure, which is "a hierarchical decomposition of the total scope of work to be carried out by the project team to accomplish the project objectives and create the required deliverables" [26]. In other words, in the project management literature, task decomposition is typically viewed as a linear, structured, top-down process for creating a hierarchy of sub-tasks. Task decomposition results in a hierarchical view of project work that includes simple operations, tasks, and sub-tasks. The process typically has a detailed series of steps focused on strategic goals, priorities, required resources, logical sequence, milestones, and eventually produces a task flowchart that shows all levels of project breakdown.

While the PMBOK guidelines are useful within general project management for their emphasis on high-level goals, priorities, resources required, milestones, and completion criteria, this top-down, linear, hierarchical process may not be optimal for data science projects in which future tasks and goals are often dependent on results of previous analyses, and cannot be precisely specified in advance.

2.2.2. Modularity and Software Engineering. In the earliest days of software development, programmers intuitively decomposed a programming task into modules, and programming was essentially a craft discipline. However, in the early 70's, Niklaus Wirth described a decomposition procedure that aimed to identify modules of a solution whose work could proceed independently of other work [27] and was used to describe the design of systems [28]. Later, modularity was noted as an approach in the design of software products [29]. Over time, more formal rules evolved (e.g. structured programming, object-oriented analysis and design, etc.) and guidelines for decomposition became formalized and incorporated into programming tools, such as interactive development environments.

This focus on work that can be done independently of other work is a central principle in the team-based task decomposition that has now been integrated into software development coding tools. In fact, the notion of modularity is central in the design and production of software artifacts, especially for large and complex projects [5]. In other words, for information system / computer science development efforts, the widespread adoption of object oriented languages and the diffusion of component based development as well other popular trends in software engineering means that software developers are exposed to modular thinking throughout their post-secondary education as well as after graduation, when they then use those concepts within a software development context. This is very different that data scientists, who are often not exposed to this task modularity during their data science education.

2.3. Data Science Education

Perhaps because it is a new domain, we note that there has been little focus on what skills data science students should gain that could improve their ability to execute data science team projects. For example, there has not been significant discussion of the challenges that students might encounter when they are doing a data science project.

However, there has been some research on students working on data science projects within a course context. For example, one study explored how student teams worked on data science projects using different methodologies [30] and a different study discussed a project-focused data science course [31], but the focus of that study was on the viability of using real world projects not on how the team actually worked together. In fact, neither of these efforts focused on the task modularity within a project, and no research has been identified that focuses on this topic within a data science context.

There has also been some research published on the slightly more general topic of data science education. For example, some have focused on designing a data science curriculum [32,33] and others have focused on the overall design of an introductory data science course [34, 35] and yet another focused on data science pair-programming [36]. In the end, it is not surprising that it has been noted that there has been little research reported on how to educate data science students [37].

2.4 Kanban in the Classroom

Kanban was created for lean manufacturing, but has been adopted across a number of domains, including software development [38]. A key aspect of this methodology is the Kanban board, where the work in progress can easily be seen and tracked [39]. Specifically, the phases of a project are shown as

columns on a Kanban board. Within each phase, there is typically a defined maximum number of work-in-progress tasks. Using this framework, the team defines a prioritized list of what to do. Then, based on the number of allowed simultaneous tasks at each phase (column on the board), a task flows through the defined process. Limiting the number of tasks within any one phase, known as limiting work-in-progress (WIP), helps to ensure that the team minimizes bottlenecks [40] and wasted effort and also enables agility, in that the team can quickly reprioritize tasks that have been proposed but not started.

More specifically, the following are the key Kanban principles, based on Anderson's description of the Kanban methodology [39]. First, *visualize the workflow* refers to splitting the work into pieces; writing each task on a card, putting that card on the board and using named columns to illustrate where each item is in the workflow. Making the work visible—along with blockers, bottlenecks and queues—is believed to lead to increased communication and collaboration.

The other key aspect of Kanban is to *limit the work-in-progress* (WIP). By limiting how much unfinished work is in progress, the team can hopefully reduce the time it takes an item to travel through the Kanban system (i.e., for the task to be completed). This limiting of WIP can also help avoid problems caused by task switching. The idea is that by using work-in-process limits the team can smooth the flow of work and make sure the team is focused on getting work completed as well as collect metrics to analyze flow.

There is growing research demonstrating the benefits when student teams use a Kanban approach. For example, it has been empirically shown that Kanban provides increased motivation and project activity control [41]. In addition, a case study of students using the Kanban methodology found that the students who applied the Kanban principles in their project work perceived an increase in outcome success [40]. It was also found that the majority of the students expressed positive views about Kanban in their project work and appreciated its value as part of their university education. Others have also reported on the benefits of using a Kanban based methodology for capstone projects [42].

More generally, a recent study statistically compared the effectiveness of the Scrum and Kanban methods for software development projects [43] and found that both Scrum and Kanban lead to the development of successful projects, but that the Kanban method was better than the Scrum method.

In terms of research on Kanban practices within a data science classroom context, in an experiment comparing several different process methodologies for use within a group data science project, it was noted that a Scrum methodology performed the worst due to the challenge of students being able to estimate task duration and Kanban performed the best, in part due to not requiring explicit task estimation [10].

Hence, with this background in mind, our research focused on the task modularity skills and capabilities of data science students when using the Kanban methodology.

3. Methodology

Our study examines the ability of data science students to use task modularity when using a Kanban process on a group project. In this section, we first describe the context and class environment and then review the Kanban process used within the course.

3.1. Study Context / Environment

152 students in a graduate level introduction to data science class were put into teams to work on a semester long project. All students received the same large lecture instruction as well as weekly time in a smaller lab section. There were 7 lab sections in the course. There were four section instructors, with each instructor teaching one, two or three sections. Students were randomly assigned to teams, which were comprised of four to six students per team and all team members were from the same section. In total, there were 31 teams in our study.

While most of the students were graduate information system students, 13 percent were in other graduate programs, mainly business administration or public policy. The class had a broad spectrum of student undergraduate majors, including fields such as information technology, engineering, and business.

The mixed method study explored how the Kanban process methodology affected modular thinking and task modularity across the teams via three complementary approaches. First, we explored the task modularity of the actual student projects. Second, we also surveyed the students on their perceptions of how the methodology encouraged task modularity. Finally, we augmented this data with semi-structured interviews of the section instructors.

3.2. Kanban Process Description

At the start of the project, the students received an explanation of the Kanban process to be used during their team project. This explanation typically took about an hour of class time (including student Q&A). Throughout the semester, the teams received feedback on their use of the methodology from their instructor.

The Kanban project methodology was based on Kanban pipeline process management described by Anderson [39] and described in the previous section. The teams were given the freedom to define the columns (the different phases of the project) on their board that they thought were most useful, however a default configuration was suggested that had four columns: "to do", "doing", "validating" and "done". To help define and track work, the teams used trello (www.trello.com), which is an online web-based tool for visualizing a board.

From a task perspective, each team was asked to define what they wanted to investigate (i.e., tasks such as "link weather data to our previously collected data"). These ideas were all listed (in a prioritized order) in their "to do" column. Then, as space permitted (based on the number of allowed simultaneous tasks at each step), a task was permitted to flow to the next column on the board. In other words, when a task was completed within a column, that task got moved to the next column and so on across the board until the task is completed. Each team set their own WIP limits, and the WIP limit per column was typically the number of people in the team.

As the board allowed (based on the work-in-progress limits), new tasks could be started. Each team also decided on the size of the "chunks of work" (tasks to be done). However, it was explained to the teams that the smaller / more detailed the task, the easier it would be for the team to understand potential bottlenecks.

Hence, the process to do a data science project could be thought of as a pipeline with requests entering one end and improved data insight coming out the other end. The team worked through the project pipeline throughout the project with no specific deadlines for interim deliverables. The goal was to make sure, at the end of the semester, that there was not a lot of time spent on an effort that did not complete (better to get a fewer number of tasks all the way through the pipeline). In summary, the key Kanban-based principles, based on Anderson's description of the Kanban methodology, were explained to the students, and included concepts such as visualizing the workflow, limiting work-in-progress and focusing on the flow of the Kanban board.

3.3. Measuring Task Modularity

One approach to measure the task modularity for a project that uses Kanban is to evaluate the Kanban board used that team. Specifically, we explored each board to see if each of the tasks were defined in a modular fashion.

Specifically, each team's Kanban board was evaluated twice. The first board evaluation was three weeks after the project started, and the second evaluation was one month later, after instructor coaching on task modularity.

Two independent coders evaluated each of the 31 Kanban boards to determine each board's task modularity. The coders were experienced data scientists and were provided high level guidelines to evaluate the tasks (e.g., evaluate the required time to complete the task), as well as some specific criteria to help evaluate the tasks (e.g., did the task have a clearly defined goal). However, it was also recognized that determining what was a "good modular task" required some human judgment. Perhaps due to this required judgment, after training, the coders agreed on 85% of the coding decisions. Disagreements were discussed and agreed upon to create a final coded data set.

In general, for Poor/low modularity, the tasks were either too big or too small (in terms of taking a reasonable amount of time to complete), and/or the input/outputs were not clearly defined (so, when reading the task, the goals of the task were difficult to know). An example of tasks with low modularity include "identify drivers for customer satisfaction", which was the overall goal of the project and so was too vague of a task, and "compute average customer satisfaction", which based on the team's current status, would have taken just a couple of minutes to complete.

For good/high modularity, the tasks were appropriate in terms of the expected scope of the task (ex. it would take a reasonable amount of time to complete). In addition, the input and outputs of that task were clear. An example of a task with good modularity was "generate a linear model for customer satisfaction based on our 10 identified target variables".

4. Findings

4.1. Task Modularity for the Project

As shown in Table 1, for the initial analysis, of the 31 teams analyzed, only two teams (6%) had good task modularity and 19 (61%) were rated as poor. While there was some improvement after instructor coaching, there were still only six teams (19%) having good task modularity. Taken together, we can note that many of the teams had significant challenges creating modular tasks.

Table 1: Task and Project level modularity

	Initial	Follow-up
Poor	19 (61%)	11 (35%)
Fair	10 (32%)	14 (45%)
Good	2 (6%)	6 (19%)

Furthermore, the instructor's perceptions matched the findings of the analysis of the Kanban boards – that while there was an improvement from previous semesters, the students still struggled with task modularity. One interesting finding when getting feedback from the instructors was that the boards provided a vehicle in which the instructors were able to easily provide structured feedback on a student's (or the student team's) task modularity. Hence, within this context, instructors thought that the use of this process was helpful, in that students were able to learn via the feedback (coaching) that the instructors were able to provide throughout the semester.

4.2. Student Perceptions

At the end of the project, each student was given a survey to complete. Out of the 152 students, 134 responded to the end of semester survey (88% response rate).

One question asked a 5-point Likert-type question (the extent to which they agreed or disagreed with the following statement: "I think using a Kanban board improved the task modularity of our project") and 84% of the students agreed or strongly agreed with the statement. Hence, the students thought that the methodology improved their task modularity.

In addition, to better understand the students' thinking with respect to modularity, they were also asked an open-ended question. Specifically, the survey asked, "Please provide some context / information on your answer to the previous question relating to if you thought using a Kanban board improved the modularity of the project". The answers collected were analyzed through an iterative process of item surfacing, refinement and regrouping.

4.2.1 Kanban Helping with Task Modularity. An analysis of the comments showed that most students (76%) understood at least some of the benefits of task modularity when executing their projects. Some of the student comments were quite insightful, and clearly articulated benefits associated with breaking a complex project down into smaller chunks. Specifically, we identified four key themes noted by the students, each of which is described below.

Smaller tasks improve understanding of the overall project - Students were able to articulate one the key benefits of modularity, that is, task decomposition via comments such as:

"I think that the usage of trello boards really helped to divide the entire project into smaller tasks"

"We used Trello to detail each and every task we worked on. For example, a task like visualization was split into two people where one would handle bar charts and maps and the other would handle scatter plots and heat maps. Similarly, models were also split."

"Everyone uploaded tasks individually which they thought were important and later we could discuss them and find out which one is actually needed"

Task modularity improves overall project tracking - A different theme noted by students was that having task modularity enabled them to more easily track progress of their project. This was exemplified by comments such as:

"It helped us in having a look at our progress with the tasks and the pending tasks"

"We could focus more efficiently on the tasks in the to do section"

"Moving completed tasks to completed section and proposed tasks to the to do list helped us to focus on a few tasks at a time in an efficient manner"

"This helped us understand how well we were paced with the project"

Modular tasks facilitate distributing the work - Students also realized that by decomposing the project into modular components, they could more easily divide the work across the team as noted via the following student comments:

"Having individual tasks also helped us in the distribution of work among the team members"

"This division of tasks also helped in distribution of the tasks among individual team members"

Task modularity improves individual task tracking – The final theme noted by students was that using a modular approach also enabled the teams to more easily track the progress of individual tasks, exemplified by comments such as:

"Moving the finished tasks to the completed section also helped us keep track of the status of the pending work"

"Each team member could monitor the other persons work right to the very detail"

"We could break up the project into multiple tasks and tackle them one at a time"

4.2.2 Kanban Not Helping with Task Modularity. However, the analysis of student comments also showed that a number of students found it difficult to effectively modularize their project tasks, which was consistent with what was noted by the analysis of the actual Kanban boards (that were created within Trello). In reviewing student comments, 22% of the students articulated challenges in modularizing their work. These comments suggest that at least some students were willing to articulate the problems they experienced when trying to break down a complex project into workable chunks. Our analysis of the Kanban boards, described in the previous section, indicates that these problems were probably more widespread than even these comments indicate.

Specifically, in reviewing the student comments, we noted two key themes relating to the Kanban process not being helpful with respect to modularity.

Hard to divide complex tasks into chunks of appropriate size/scope – Knowing that it is useful to create subtasks is not the same as being able to create subtasks that are useful. This was a key challenge noted when evaluating the Kanban boards, and was also noted by several students, for example:

"Our team was unsure on how to break down the more complex tasks, which led us to having a team member focus on one modeling solution at a time in a silo, rather than having it broken out more incrementally"

"While trying to understand the data science concepts at a basic level, I don't think we internalized how to break them down into modules or small tasks"

"It was sometimes difficult to divide your task into smaller chunks due to lack of proper communication"

Similarly, a related challenge in decomposing the project was to have a good grasp on how large each module should be – being too large or too small was noted as being problematic:

"It can help us to divide big tasks but we still face some problems like how to divide tasks equally"

"There were times when we tried to divide up a task into pieces just so that people could have a task to move but in the end, we would either complete the task together or individually before comparing results"

"Division of work looked fine on trello board, but implementation of the tasks on separate machines and then combining them was a bigger task"

Process was confusing – In addition, some students were confused with respect how to use the process. This could have been due to the combination of working on a difficult project, working with a process methodology that was new to them and trying to create modular components, which was also a new concept. Additional education and training might address these issues, which were typically noted in very general terms, such as:

"[the process] actually made this process more confusing"

"[the process] was just a burden as we did not need that as opposed to meeting regularly as a team"

"Our group was also very small, the task assignment process is usually only helpful when the group is 10+"

5. Discussion

From our analysis of the project boards, it is clear that decomposing large and complex data science tasks into discrete, re-combinable subtasks was a challenge for most of the data science students. While some teams did improve from their initial efforts, even after gaining comfort in the use of the Kanban process, task modularity in most teams was not as

good as one would hope to see, since only 19% of the teams achieved good task modularity. Hence, the evidence from our study makes it clear that decomposing large and complex tasks into discrete, re-combinable subtasks was still a challenge for many data science teams.

However, it is also clear from our study that students did recognize the rationale for, and benefits of, effective decomposition of complex project tasks. This suggests that the challenge is not student motivation, but rather, that creating good modular sub-tasks is difficult and that the students need more than the training and coaching provided within the Kanban context. In other words, the fact that there were still many teams who were unable to decompose effectively suggests that opportunities remain in terms of how to best enable data science teams to effectively achieve task modularity.

As a first step towards the goal of helping students improve task modularity, we provide some potential best practices for task decomposition. Specifically, based on our observations, we developed six guidelines that might help students improve their ability to effectively decompose data science projects into workable sub-tasks. These approaches are often complimentary in nature, in that they could be used in conjunction with each other to help in the process of task decomposition.

Note that these six practices could be explored, refined and elaborated upon in future research investigating data science task modularity.

Have a specific and concise task title - Since the task description, or title, is often how people refer to the task, it is important to have a well-defined task name. Titles should also be short and focused. It is helpful to start the title with a verb. The title is the first step to ensure everyone understands what will be done within the task.

Have a well-defined goal - Ensure that the task has a clearly defined goal. In other words, what is the purpose of the task/module, and why is it important to complete the task? How does this module help create actionable insight? It is also important that others can easily understand the goal of the task, which should be suggested by the task title, but elaborated as needed to ensure a consistent view across the team of the goal of the task.

Define task inputs and outputs - In addition to having a clearly defined goal, it is also important to clearly articulate what inputs the module needs (ex. data attributes columns within a data file which might have been generated as an output from a previous task, such as data cleaning). It is also important to clearly define the outputs that will be generated from the module, which might be cleaned data sets, visualizations or model output. By having clearly defined inputs and outputs, teams can achieve high cohesion and low coupling.

Ensure reasonable task duration - Care should be given to the duration of the task. A task that will take one month to complete will involve significant work, and will lessen the impact of a modular approach. Similarly, it is possible to define tasks with durations that are too short, leading to a focus on the trivial and excessive task management overhead. The study suggests that one of the hardest challenges for students was to understand how to best decide on the granularity of the task. In essence, the challenge is to ensure the task is "not too big, but also, not too small". For example, one could try to use a time-based approach, where one tries to determine the granularity of the task by suggesting task duration. However, this can be difficult to implement, since estimation of the time it takes to do a task is one of the difficulties of students when executing data science projects [10].

Ensure a logical start and end - Tasks should have a natural and logical start and finish, which could be analogous to single entry and exit in software modules. Based on this approach, a logical task can be broken into subtasks. If there are more than 7 (+/- 2) sub-tasks per phase, as suggested by PMBOK, then that task should be broken into smaller tasks. One keeps breaking down tasks until one defines a small set of subtasks. One risk in this approach is that one might create too many small tasks.

Define accountability and responsibility - Every task should have a clear-cut person (or team) working on the task, and there should be a clearly defined owner of that task. In addition, every task should have clearly defined completion criteria. This approach helps to ensure clearly defined tasks that others can easily understand.

6. Conclusion

Task modularity within a data science context is a new area that has not previously been studied. To address our first research question, we note that an analysis of the student's initial attempt at task modularity demonstrates that data science students do not naturally apply task modularity to their projects. To address our second research question, we note that in general, the students still had difficulty achieving task decomposition and task modularity, even after exposure to the Kanban methodology and task modularity coaching support by the instructors.

Some of the challenges in achieving task modularity were that it was difficult to divide

complex tasks, and many of the team's tasks were perceived to be complex. In addition, those complex tasks were difficult to size/scope, so they often were either very large or very small.

At a higher level, we note that while computer science/information system programs typically teach modularity (e.g., the decomposition process and abstraction along with topics such as patterns and components), to date, there does not seem to be a corresponding approach of how to teach modularity within a data science context. Our rules of thumb do not fully address this gap and there remains a need to improve these potential best practices and more importantly, identify a corresponding model to that used for computer science / information system students, to teach modularity to data science students.

6.1. Potential Next Steps

First, we note that the evolution within the software development domain is perhaps analogous to the task decomposition challenges facing data science students and practitioners today, where a tool-focused approach might be applicable within a data science context. However, data science is typically viewed via a data flow construct, not an object-oriented approach. Hence, future work needs to investigate the applicability of using a tool-based approach for modular data science efforts, but with the acknowledgement of how data scientists typically work.

Specifically, related to exploring tools to support modularity, one could explore group coordination and decomposition tools that could be integrated with code modules. One such example of a group coordination tool is Trello (www.trello.com), which was used in this study and provides boards to make task decomposition visible. Future work could explore how such a team-based tool could be integrated within a code-based modular development environment, which could make task decomposition more focused and hence easier for data scientists and data science students.

6.2. Limitations

This mixed method study had several limitations, which additional research could address. For example, this effort leveraged graduate students. Junior data science professionals or undergraduate students might yield different results. Related to this, most of the students were information system students and it is possible participants with a different background, especially more computer science focused students, via their significant object-oriented programming experience, might have a better approach to data science task modularity. Furthermore, the type of data science project might have impacted our results, and so, additional case studies could be done to identify if the type of project impacts a student's ability to achieve task modularity.

7. References

[1] M. Das, R. Cui, D. R. Campbell, G. Agrawal, and R. Ramnath, Towards methods for systematic research on big data, in *Big Data (Big Data), IEEE International Conference on*, pp. 2072-2081, 2015.

[2] J. Saltz and I. Shamshurin, Big Data Team Process Methodology: A Literature Review and the Identification of Critical Factors for a Project's Success, in *Big Data (Big Data), IEEE International Conference on,* 2016.

[3] C. Baldwin and K. Clark, Modularity in the Design of Complex Engineering Systems, Working Paper, *Harvard Business School*, Boston, MA. 2004.

[4] C. Mattmann, (2013). Computing: A vision for data science. *Nature*, *493*(7433), 473-475, 2013.

[5] Y. Yeo, J., Hahn, The Role of Project Modularity in Information Systems Development, *35th International Conference on Information Systems*, 2014.

[6] A. Salonen, R. Rajala and A. Virtanen, Leveraging the benefits of modularity in the provision of integrated solutions: A strategic learning perspective. *Industrial Marketing Management*, 2017.

[7] D. Sturtevant, D. Modular Architectures Make You Agile in the Long Run. *IEEE Software*, (1), 104-108, 2017.

[8] M. Schilling. Toward a general modular systems theory and its application to interfirm product modularity. *Academy of Management Review*, 25(2), 312–334, 2000.

[9] R Core Team, R: A Language and Environment for Statistical Computing, *R Foundation for Statistical Computing,* available at https://www.R-project.org/, 2016.

[10] J. Saltz, I. Shamshurin and K. Crowston, Comparing Data Science Project Management Methodologies via a Controlled Experiment. in *Hawaii International Conference on System Sciences (HICSS)*, 2017.

[11] M. Vanauer, C. Bohle and B. Hellingrath, Guiding the Introduction of Big Data in Organizations: A Methodology with Business-and Data-Driven Ideation and Enterprise Architecture Management-Based Implementation, in *Hawaii International Conference on System Sciences (HICSS)*, 2015.

[12] A. Bhardwaj, S. Bhattacherjee, A. Chavan, A. Deshpande, A. Elmore, S. Madden and A. Parameswaran, DataHub: Collaborative Data Science & Dataset Version Management at Scale, Bi*ennial Conference on Innovative Data Systems Research (CIDR)*, 2015.

[13] J. Gao A. Koronios and S. Selle, Towards A Process View on Critical Success Factors in Big Data

Analytics Projects, *21st Americas' Conference on Information Systems, 2015*.
[14] N. Grady, M. Underwood, A. Roy and W. Chang, Big Data: Challenges, practices and technologies: NIST Big Data Public Working Group workshop at IEEE Big Data, in *Big Data (Big Data), IEEE International Conference on*, pp. 11-15: IEEE, 2014.
[15] N. Chandler and T. Oestreich, *Use analytic business processes to drive business performance*, ed: Gartner, 2015.
[16] H. Chen, R. Kazman and F. Matthes, Demystifying big data adoption: Beyond IT fashion and relative advantage. In *Proc. DIGIT*, 2015.
[17] J. Espinosa and F. Armour, The Big Data Analytics Gold Rush: A Research Framework for Coordination and Governance, in *49th Hawaii International Conference on System Sciences (HICSS)*, pp. 1112-1121: IEEE, 2016.
[18] H. Chen, R. Kazman and S. Haziyev, Agile Big Data Analytics for Web-based Systems: An Architecturecentric Approach, *IEEE Transactions on Big Data*, 2016.
[19] V. Dhar, Data science and prediction, *Communications of the ACM*, vol. 56, no. 12, pp. 64-73, 2013.
[20] J. Saltz, The need for new processes, methodologies and tools to support big data teams and improve big data project effectiveness, in *Big Data (Big Data), IEEE International Conference on*, 2015.
[21] C.Y. Baldwin and K.B. Clark, The Value and Costs of Modularity, Working Paper, *Harvard Business School*, Boston, MA, 2001.
[22] S. Brusoni, L. Marengo, A. Prencipe and M. Valente, The Value and Costs of Modularity: A Problem-Solving Perspective, *European Management Review* (4:2), pp. 121-132, 2007.
[23] R.N. Langlois, Modularity in Technology and Organization, *Journal of Economic Behavior & Organization* (49:1), pp. 19-37, 2002.
[24] C.Y. Baldwin and K.B. Clark, Managing in an age of modularity. *Harvard Business Review* 84–93 September-October, 1997.
[25] PMBOK Guide: A Guide to the Project Management Body of Knowledge, *Project Management Institute (PMI),* Pennsylvania, 2004.
[26] Work Breakdown Structure, in *Wikipedia*, 2018, DOI: en.wikipedia.org/wiki/Work_breakdown_structure
[27] F. P. Brooks, Mythical Man-Month. *Datamation*, 20(12), 44-52, 1974.
[28] D. Parnas, On the Criteria to Be Used in Decomposing Systems Into Modules, *Communications of the ACM*, 15, 1053–1058, 1972.
[29] C. Szyperski, Independently Extensible Systems—Software Engineering Potential and Challenges, *Australian Computer Science Communications*, 18, 203–212, 1996.
[30] J. Saltz, R. Heckman and I. Shamshurin, Exploring How Different Project Management Methodologies Impact Data Science Students, In *Proceedings of the 25th European Conference on Information Systems (ECIS)*, 2017.
[31] J. Saltz and R. Heckman, Big Data Science Education: A case study of a Project-Focused Introductory Course, *Themes in Science and Technology Education, Special issue on Big Data in Education*, 8(2), 2016.
[32] B. Ramamurthy, A Practical and Sustainable Model for Learning and Teaching Data Science'. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*: ACM, 169-174, 2016.
[33] P. Anderson, J. Bowring, R. McCauley, G. Pothering, and C. Starr, An undergraduate degree in data science: curriculum and a decade of implementation experience. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 145-150). ACM, 2014.
[34] Y. Gil, Teaching parallelism without programming: a data science curriculum for non-CS students. *Proceedings of the Workshop on Education for High-Performance Computing*: IEEE Press, 42-48, 2014.
[35] R. Brunner and E. Kim, Teaching Data Science, *Procedia Computer Science,* 80, pp. 1947-1956, 2016.
[36] J. Saltz and I. Shamshurin, Does Pair Programming work in a Data Science Context: An Initial Case Study, in *Big Data (Big Data), 2017 IEEE International Conference on*, 2017.
[37] M. Mellody, Training Students to Extract Value From Big Data, Summary of a Workshop, *The National Academies Press*, Washington DC, 2014.
[38] M. Ahmad, J. Markkula and M. Oivo, Kanban in software development: A systematic literature review, *Software Engineering and Advanced Applications (SEAA), 39th EUROMICRO Conference on*, pp. 9-16: IEEE, 2013.
[39] D.J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business. *Blue Hole Press*, 2010.
[40] M. O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review," in *Software Engineering and Advanced Applications (SEAA), 39th EUROMICRO Conference* on, pp. 9-16: IEEE, 2013.
[41] M. Ikonen, E. Pirinen, F. Fagerholm, P. Kettunen and P. Abrahamsson, On the impact of Kanban on software project work: An empirical case study investigation. *In Engineering of Complex Computer Systems (ICECCS), 16th IEEE International Conference on* (pp. 305-314). IEEE, 2011.
[42] A. Neyem, J. Diaz-Mosquera, J. Munoz-Gama and J. Navon, Understanding Student Interactions in Capstone Courses to Improve Learning Experiences. *In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 2017.
[43] H. Lei, F. Ganjeizadeh, P.K. Jayachandran and P. Ozcan, A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing,* 43, 59-67, 2017.