Environmental scientists, land managers, and policy actors are increasingly presented with high-stakes high-uncertainty problems stemming from human-ecosystem interactions. To help address these problems, scientists frequently use models that produce enormous geospatial and temporal datasets that are constantly modified and often seek input from communities outside their discipline. To assist scientists—as well as others who interface with policymakers—in generating insights from this complex and changing data, this research examines a co-production effort where ecologists, environmental scientists, computer scientists, software engineers, and social scientists collaborate on the development of domain-specific software as part of the Visualization of Terrestrial and Aquatic Systems (VISTAS) project. Findings from this case study suggest that visualization is critical for communicating science to both experts and non-experts, and integral to many aspects of the science production pipeline and policymaking process, including data exploration and model validation. Additionally, there is evidence among our collaborators that this software co-production process not only resulted in useable and useful application features for visualization and data analytics, but also influenced the way that scientists approach their research activities. This research has potential implications for both the production of domain-specific
software and the co-production of scientific knowledge, highlighting the challenges of embarking on a software co-production effort that spans multiple years with changing project personnel and shifting priorities. Recommendations for policies on how to promote and support such research activities both within this domain context and other institutional contexts are also discussed.
Co-producing Software for Environmental and Ecological Visualization

by

Chad Zanocco

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented February 8, 2019
Commencement June 2019
Doctor of Philosophy dissertation of Chad Zanocco presented on February 8, 2019

APPROVED:

Major Professor, representing Public Policy

Director of the School of Public Policy

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Chad Zanocco, Author
ACKNOWLEDGEMENTS

The author expresses sincere appreciation to all the faculty, researchers, and collaborators involved with the VISTAS project for their invaluable contributions of both time and expertise. I am also indebted to my family, friends, and colleagues for supporting and encouraging me throughout this process. I would like to thank the National Science Foundation (BIO/DBI 1062572) and the School of Public Policy at Oregon State University for their generous support of this research.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Literature review</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Big data: Prospects and Pitfalls</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Big data characteristics</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2 Big data in ecology and environmental science</td>
<td>8</td>
</tr>
<tr>
<td>2.1.3 Big data visualization</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Software design and development</td>
<td>12</td>
</tr>
<tr>
<td>2.2.1 Plan-driven software development</td>
<td>13</td>
</tr>
<tr>
<td>2.2.2 User-centered agile software development</td>
<td>14</td>
</tr>
<tr>
<td>2.2.3 Design Frameworks for Visualization Software</td>
<td>17</td>
</tr>
<tr>
<td>2.3 Co-production of knowledge</td>
<td>20</td>
</tr>
<tr>
<td>2.3.1 Co-production for ecology and environmental science</td>
<td>24</td>
</tr>
<tr>
<td>3 Methods</td>
<td>28</td>
</tr>
<tr>
<td>3.1 Research Design</td>
<td>29</td>
</tr>
<tr>
<td>3.1.1 Longitudinal case study designs</td>
<td>29</td>
</tr>
<tr>
<td>3.1.2 Pre-post case study design and the VISTAS project</td>
<td>30</td>
</tr>
<tr>
<td>3.1.3 Group context</td>
<td>31</td>
</tr>
<tr>
<td>3.2 Data Collection</td>
<td>34</td>
</tr>
<tr>
<td>3.2.1 Recruitment and interview context</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Data collected from project materials and participant interactions</td>
<td>37</td>
</tr>
<tr>
<td>3.3 Data analysis: qualitative coding</td>
<td>39</td>
</tr>
<tr>
<td>3.3.1 Codebook development</td>
<td>39</td>
</tr>
<tr>
<td>3.3.2 Overview of the codebook</td>
<td>42</td>
</tr>
<tr>
<td>3.3.3 Data Analysis Strategy</td>
<td>47</td>
</tr>
<tr>
<td>3.4 Validity and Ethical Issues</td>
<td>47</td>
</tr>
<tr>
<td>3.4.1 My embeddedness with the VISTAS project</td>
<td>47</td>
</tr>
<tr>
<td>4 Findings</td>
<td>51</td>
</tr>
<tr>
<td>4.1 Pre-period</td>
<td>51</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1 Visualization</td>
<td>51</td>
</tr>
<tr>
<td>4.1.2 Exploration</td>
<td>60</td>
</tr>
<tr>
<td>4.1.3 Communication</td>
<td>70</td>
</tr>
<tr>
<td>4.1.4 Co-production</td>
<td>79</td>
</tr>
<tr>
<td>4.2 Development period</td>
<td>81</td>
</tr>
<tr>
<td>4.2.1 A brief history of the development period</td>
<td>82</td>
</tr>
<tr>
<td>4.2.2 Co-production</td>
<td>84</td>
</tr>
<tr>
<td>4.3 Post-period</td>
<td>100</td>
</tr>
<tr>
<td>4.3.1 Co-production</td>
<td>100</td>
</tr>
<tr>
<td>5 Discussion</td>
<td>125</td>
</tr>
<tr>
<td>5.1 Visualization needs and software design</td>
<td>125</td>
</tr>
<tr>
<td>5.1.1 Challenges with data</td>
<td>125</td>
</tr>
<tr>
<td>5.1.2 Challenges with visualization</td>
<td>127</td>
</tr>
<tr>
<td>5.2 Development of visualization software</td>
<td>131</td>
</tr>
<tr>
<td>5.3 Co-production</td>
<td>135</td>
</tr>
<tr>
<td>5.3.1 Inputs</td>
<td>135</td>
</tr>
<tr>
<td>5.3.2 Co-production development and process</td>
<td>137</td>
</tr>
<tr>
<td>5.3.3 Outputs</td>
<td>139</td>
</tr>
<tr>
<td>5.3.4 Relevance of co-production and changes in practice</td>
<td>140</td>
</tr>
<tr>
<td>5.4 Policy and Practice</td>
<td>142</td>
</tr>
<tr>
<td>5.4.1 Software design for domain-specific problems</td>
<td>142</td>
</tr>
<tr>
<td>5.4.2 Software development process</td>
<td>143</td>
</tr>
<tr>
<td>5.4.3 Facilitating software co-production at institutional levels</td>
<td>145</td>
</tr>
<tr>
<td>5.4.4 Co-production as an approach for knowledge creation</td>
<td>147</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>149</td>
</tr>
<tr>
<td>7 Bibliography</td>
<td>155</td>
</tr>
<tr>
<td>8 Appendices</td>
<td>171</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>The four paradigms of science: empirical, theoretical, computational, and data driven. Adapted from Agrawal and Choudhary (2016).</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Integrating Agile Methods and Human Centered Design. Adapted from Silva et al. (2011).</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Four levels of design with threats to validity. Adapted from Munzer (2014).</td>
<td>19</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Logical co-production model framework adapted from Djenontin and Meadow (2018).</td>
<td>41</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Normal Q-Q plot</td>
<td>89</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Residuals vs. leverage plot</td>
<td>89</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Residuals vs. fitted plot</td>
<td>90</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Scatter plot with best fit line</td>
<td>90</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Scale-location plot</td>
<td>91</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Scatterplot with alpha = 1</td>
<td>92</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Scatterplot with alpha = 0.1</td>
<td>92</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Linear regression output from R</td>
<td>93</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Linear regression scatterplot (left) of visualized landscape data (right)</td>
<td>95</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Linear regression heatmap (left) of visualized landscape data (right)</td>
<td>96</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Early development prototype of PCA visual display</td>
<td>97</td>
</tr>
<tr>
<td>Figure 16</td>
<td>PCA with scatterplot display (left) of visualized landscape data (right)</td>
<td>98</td>
</tr>
<tr>
<td>Figure 17</td>
<td>PCA with heatmap display (left) of visualized landscape data (right)</td>
<td>98</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Table 1: Proposed timeline in early 2017.</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Table 2: Finalized timeline in mid-2018</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Table 3: Interview summary information</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>Table 4: Interview schedule</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>Table 5: Finalized codebook</td>
<td>42</td>
<td></td>
</tr>
</tbody>
</table>
### LIST OF APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Appendix A: Pre-development interview protocol</td>
<td>171</td>
</tr>
<tr>
<td>8.2 Appendix B: Post-development interview protocol</td>
<td>173</td>
</tr>
</tbody>
</table>
1 Introduction

Society is continually confronted with “wicked problems,” which are identified as intractable problems that are challenging to address due to their complexity and open-endedness (Head 2008, p. 101). Indeed, many of the wicked problems facing society are ecological problems, and are problems often driven by human-environment interactions (Balint 2011). Since solving such problems is particularly challenging using a more traditional (i.e. normal) science approach, new approaches and techniques are needed for managers and others seeking to effectively confront and address such problems. Lessons for addressing wicked-type problems in prior scholarship include: “(1) stop looking for the perfect solution; (2) seek instead a satisficing response; and (3) consider applying the iterative, analytic, adaptive, participatory process” (Balint 2011, p. 207). It is with these lessons in mind that we embarked on a collaborative, interdisciplinary effort to develop technologies that help environmental scientists, ecologists, and ultimately managers and stakeholders to make more informed policy decisions in wicked problem contexts.

Recent technological innovations in computing provide new opportunities for understanding and addressing complex ecological and environmental problems, leading to advancements in both the way that data\(^1\) is collected and processed (e.g., computer processing speed, remote sensing technologies, storage), and how this data is interpreted and applied (e.g., statistical software, landscape modeling, visualization). Using emerging technologies that can more fully capture the behavior of complex systems, researchers

---

\(^1\) Throughout this study, the term *data* is used as both a singular and plural noun. This dual usage is accepted by many style guides and dictionaries (Izzo 2012).
now have powerful tools at their disposal which they can use to characterize and model ecological and environmental processes. However, for many in this domain, these improvements, while providing new sources of actionable information, have also resulted in troves of data. This flood of data, produced through methods such as computer-based modeling and *in situ* sensor returns, present new obstacles for interpretation, and, ultimately, applications for informing management decisions and policymaking. While these technological advances have contributed to both the speed and volume of data generation, making sense of these ever-expanding data streams remains an on-going challenge. Developing and designing systems to better understand the volume of information generated by these models is a constantly evolving endeavor and represents a substantial challenge for ecological and environmental science research, for which this research effort attempts to address just a part.

With this broader effort in mind, this dissertation explores ways that large, spatially and temporally complex datasets produced through both observed and modeled techniques can be better understood by combining computer-based technologies, such as topographic visualization and data analytics, with methods from the social sciences designed to understand and confront complex problems of high societal importance. Specifically, we consider experiences from the VISTAS (Visualization of Terrestrial and Aquatic Systems) project, an on-going effort to produce useful and useable software for environmental scientists and ecologists to help them understand, and ultimately address, problem contexts that often manifest in ways that have varying degrees of wickedness.

One proposed process for confronting wicked-type problems is co-production of knowledge, a burgeoning research approach frequently applied to complex environmental
problem domains, often of critical societal concern. In this study, I explore the extension of this co-production approach to the development of software for applications in environmental science and ecology. While a co-production of knowledge approach has been applied in web-based application development for land managers (e.g., Zanocco et al. 2018), scholarship that considers co-production activities involving software development are uncommon. In this research, I attempt to flesh out the theoretical and methodical dimensions surrounding software co-production by embedding myself in a multi-disciplinary, collaborative effort that develops visualization software designed to help scientists understand spatially and temporally complex information at landscape scales. I also explore the scientific output, which, in the context of this project, includes visualization software designed to display complex data at topographic and temporal scales. Detailed inquiry into the scientific output of co-production activities has been identified as an underdeveloped area in extant literature (Hegger and Dieperink 2015; Voorberg, Bekkers, and Tummers 2015), and the research design of this project presents unique opportunities for investigating the resultant knowledge production output of such activities.

I therefore offer the following research questions which emerged through a review of the literature and collective goals established at the beginning of the project:

1. What are the domain specific challenges associated with managing and visualizing complex environmental data? How, if possible, can such challenges be addressed through the design of domain-specific software?
2. What are the benefits and drawbacks of applying a co-production of knowledge process to the production of visualization software? What do
we discover from engaging in such a co-production effort? Is the resultant product from the co-production effort considered useful and useable by participants?

3. How does the co-production process impact participants? Do they understand or approach their work in a different way? Does it change the way they communicate with others about their work and the way the engage in the policy process?

Before discussing the VISTAS project in more detail and begin the process of addressing these research questions within the research design proposed in this study, I will first review three areas of literature that are critical for understanding the scholarly context under which this research was undertaken. Accordingly, the next section describes relevant scholarship from data science and visualization, software design and development, and knowledge co-production. These areas of scholarship are germane to the scientific domain being explored—environmental science and ecology—as well as the process of creating visualization software in settings where developer and user goals are iteratively modified and updated.
2 Literature review

Technological advances in computing, storage, data collection, information delivery, and analytical approaches has led to the promise of transformational shifts in the way we understand the world (Gandomi and Haider 2015). These advances in data science are often encapsulated by the term *big data*, and has advanced, or is expected to advance, a myriad of fields in diverse ways, impacting areas such as research programs, commerce, and governmental decision-making. While the era of big data opens new frontiers of opportunities and possibilities, it also presents new challenges. These opportunities and challenges are particularly apparent in fields such as environmental science and ecology, where explosions in the volume of modeled and sensed data have led to questions about how to interpret and understand this data, and most importantly, generate meaningful insights from the flood of newly available information (Hampton et al. 2013). In this first section of the literature review, we will review big data, big data in the domain of ecology and environmental science, and then briefly discuss the role of visualization in big data workflows.

2.1 Big data: Prospects and Pitfalls

2.1.1 Big data characteristics

Big data, contrary to what the term implies, does not only refer to the amount of data, but is more generally understood as a collection of characteristics that distinguish certain difficult to manage data from data that are easier to manage (Constantiou and Kallinkikos 2015). While there lacks a definitive consensus on what constitutes big data (Labrinidis and Jagadish 2012), many scholars tend to agree that big data spans at least
three dimensions, sometimes referred to as the “three Vs”: volume, velocity, and variety (Chen and Zhang 2014, p. 314; Laney 2001). *Volume* refers to the size and resolution of the data, *velocity* to the speed at which the data is generated and consumed, and *variety* to the diversity and range of the data. Scholars frequently update or append this list, usually to best fit their domain or problem context and include additional Vs, such as *veracity*, which refers to the trustworthiness or dependability of the data (e.g., Hashem et al. 2015; Abbasi, Sarker, and Chiang 2016), or *visualization*\(^2\), which emphasizes the importance of displaying information within big data workflows (e.g. Sagiroglu and Sinanc 2013; Li 2016).

Distinguishing characteristics are sometimes depicted as not what big data *is*, but more often what it *is not*. For example, big data cannot be easily handled using conventional data techniques, presenting formidable challenges for tasks such as data collection, management, analysis, and visualization (Chen and Zhang 2014; Gandomi and Haider 2015). Scientists may be unaware that they are confronted with big data until limitations in their usual data management and analysis procedures are identified: data collection workflows break down, insights from data cannot be generated in ways they are accustomed to, or other kinds of processing, typically applied to their data, prove insufficient. In this respect, the traditional techniques and technologies that scientists,

---

\(^2\) In the context of this study, visualization primarily refers to scientific visualization, or the visualization of information for assisting scientific inquiry (e.g., formation and testing of hypotheses). However, in practice, the term scientific visualization is frequently used to refer to abstract or realistic renderings of volumes (e.g., three dimensional or 3D visualizations) which may, or may not, have a dynamic or temporal component. Often, the term data visualization, a subset of scientific visualization, is used to describe the visual representation of information that is typically organized through a level of abstraction such as variables, categories, or attributes (Friendly 2009).
businesses, policymakers, and others often rely on for managing and understanding their data are inadequate, inefficient, and are generally mismatched when data has big data characteristics (e.g., high velocity and high volume). The emergent field of data intensive science grew from a need to confront such issues, using a collection of techniques designed to handle big data. Indeed, some have declared data driven science as a new scientific paradigm with characteristics that are distinguishable from previously identified paradigms such as empirical science, theoretical science, and computational science (Figure 1) (Chen and Zhang 2014; Agrawal and Choudhary 2016).

Figure 1: The four paradigms of science: empirical, theoretical, computational, and data driven. Adapted from Agrawal and Choudhary (2016).

Others are more skeptical about treating big data as a distinct construct, and instead believe that big data characteristics are not necessarily unique from other previously established data characteristics. Some have cautioned against the idea of big data as being transformative and transcendental, labeling this perspective as “Big data hubris” where big data, at its face, is seen a replacement, rather than supplement to, established paradigms for collecting and analyzing data, leading to false conclusions.
(Lazer et al. 2014, p. 1203). Others have identified issues of a different sort: instead of an overreliance on big data that threatens to supplant previously established paradigms, big data advances are slow to be adopted in certain fields, such as ecology and environmental sciences, seemingly running contrary to the experiences of big data science in other domains and its potential for transformational advancement (Costello et al. 2013; Hampton et al. 2013; Peters et al. 2014; Fidler et al. 2017). Regardless of whether it be low domain adoption, misinterpretation of big data patterns, or limitations in technological capacity, current impasses exist to realizing the full potential of big data in real-world applications.

2.1.2 Big data in ecology and environmental science

In ecology and the environmental sciences, datasets are increasingly growing in size and complexity. New sensing technologies, varying from terrestrial and aquatic instruments to satellite-based systems, generate massive timeseries datasets, often with spatial referencing and autonomous updating in near-real time (Hampton et al. 2013). Legacy data, collected at a time prior to the emergence of contemporary data science storage techniques, are in the process of being archived and preserved in digital formats (Peters et al. 2014). Modeled data, often produced through computer simulations, can be run across multiple landscapes, under multiple scenarios, across multiple time periods, producing enormous time-series and spatial data outputs. While the field of ecology and environmental science is confronted with the possibility of ever-expanding data sources, data science practices, such as data sharing and the re-use of data, have had slower adoption. One reason identified for this slower adoption is that ecology, as a field, lacks a culture of data curation and sharing (Hampton et al. 2013; Peters et al. 2014).
suggests that a shift toward big data science ecology may need to concurrently address this deficiency in data curation and sharing.

However, even if data is made more available, there is no guarantee that more data will necessarily lead to a data science revolution in ecology, with Peters et al. (2014) stating that “we believe that few ecologists will take advantage of these data even if the technological and cultural challenges are met” (p. 2). Recent efforts in sharing ecological and environmental datasets have addressed some of these concerns using systems designed to facilitate data sharing (Hallgren et al. 2016) and have suggested new techniques for understanding data, such as the wider use of spatially explicit modeling (Franklin et al. 2016), and the integration of machine learning algorithms into science inquiry workflows, which are proposed to help bridge the gap between more traditional hypothesis testing and techniques more suited for analyzing big data (see Knowledge, Learning, Analysis System (KLAS); Peters et al. 2014). However, big data, when available, may not necessarily represent a panacea for ecologists, with limitations in data sampling methods (e.g., variety and veracity of the data) being unable to be overcome by simply including more data in an analysis (Engemann et al. 2015).

While not all ecological and environmental data is inherently geospatial, much of the data that is applied in research activities is georeferenced, taking such forms as raster data (e.g., satellite data in a regular grid, topographic data) and vector data (e.g., river system polylines, geographic boundaries, land use designations), with data at landscape scales having discernable geographic characteristics due to spatially explicit patterns and processes (Wagner and Fortin 2005; Fortin et al. 2016). Geographic data is not unique to ecology and environmental science, with some arguing that a majority of data is
geographic in nature, or at least has some geographic component (Hahmann and Burghardt 2013). There are many examples of geospatial data collections that produce geospatial big data including NASA’s Landsat image repositories, which houses petabytes of information (Li et al. 2016) and topographic data, collected via point clouds of Lidar sensor returns, for generating digital terrain representations such as Digital Elevation Models (DEMs) (Singh et al. 2016). In addition to spatial complexity, contributing to big data issues such as data volume and data variation, ecological data is often temporally complex as well, where the time at which data is collected varies in resolution, scale, and interval (Soranno et al. 2015). The combination of spatial and temporal characteristics creates high volume data, and, combined with other big data characteristics, such as variety and velocity, typifies big data science in ecology and environmental science, as well as other dimensions of the policy process more generally (Höchtl, Parycek, and Schöllhammer 2016; Poel, Meyer, and Schroeder 2018).

2.1.3 Big data visualization

Of the additionally proposed Vs used to describe big data, visualization and visual analytics are often referenced (Yu et al. 2014; Li et al. 2016). One reason for the importance of visualization in big data science is that traditional statistical approaches do not appropriately scale because the volume of this data is too large (Keim, Huamin and Kwan-Liu 2013; Shneiderman 2014). Additionally, data presented in tabular form, while appropriate for smaller datasets, may not be appropriate for representing the structure and features of big data (Hoffer 2014). While data visualization is often seen as critical for the process of communication, data visualization takes on new roles in big data workflows, enabling the visual powers of human perception to glean insights that may be more
difficult, or not possible to achieve, using non-visual, computer-based analytical tools (Choo and Park 2013; Li et al. 2016).

While visualization is seen as an important part of the big data pipeline, appropriate tools and expertise for facilitating data visualization are identified as deficient for environmental science and ecology, with many researchers in these fields utilizing visualizations that do not depict the complex geospatial and temporal elements of their data (Winters, Lach, and Cushing 2016). In this respect, patterns that emerge from data can be missed not because they were overlooked by the researcher, but because the researcher lacked visual systems for depicting appropriate information at a landscape scale. Prior work in ecological and environmental visualization research has demonstrated that data visualizations and visual analytics can help scientists more effectively use model outputs and large data sets to communicate complex processes and phenomena occurring in their research area to managers, policymakers, scientists, and other members of the public (Winters et al. 2016).

Multiple approaches for addressing spatiotemporal complexity in big data visualizations have either been proposed or developed, including the creation of a time-cube that can be reoriented and transformed into 2D representations based on the visualization needs of the user (Bach et al. 2016). Other such visualization approaches have considered data mining techniques for filtering subsets of large datasets to then more effectively visualize these datasets, using 3D environments if appropriate (Compieta et al. 2007). In ecology and environmental sciences, many advances in spatiotemporal visualizations have been focused on understanding satellite data, which includes high volumes of spatial and temporal information. While systems have been
purpose-built to display, query, and visualize various satellite data types (Eldaway et al. 2015), other research efforts attempting to understand temporal trends and patterns in satellite returns may also include methods to visualize these trends and patterns, including the use of existing systems such as Google Earth Engine (Kennedy et al. 2018).

As demonstrated above, while software exists that can help address big data issues, including big data visualization, ecology and some sub-fields of the environmental sciences have been slow to adopt these approaches, perhaps hampering the ability of researchers in these domains to best confront wicked environmental problems, or other difficult to address complex problems that are of high societal importance (Winters et al. 2016). Winters et al. (2016) identified a need for approaches and systems that can be applied in an ecological and environmental science context, with the authors proposing the application of a co-production of knowledge framework to develop environmental software visualizations, an approach considered in other academic scholarship attempting to address similar dilemmas (Cushing, Winters, and Lach 2015, Zanocco et al. 2018).

2.2 Software design and development

This section of the literature review provides an overview of software development and design approaches and frameworks. First, a conventional approach to software development is reviewed. Next, software development approaches involving iterative developer-user interactions are explored, as well as an example of how these approaches to development are applied towards creating software. Lastly, a framework specific to the design of domain specific software is discussed, the nested blocks and guidelines model, which utilizes user-centered approaches in the creation of visualizations.
2.2.1 Plan-driven software development

A plan-driven approach to software development responds to a need for software products to pass through quality phases including “engineering, design and implementation, testing, release, and maintenance” and is often associated with more conventional approaches to software development (Petersen and Wohlin 2010, p. 660). In a plan-driven approach, the properties of the final product are known before the development begins, allowing for a process that can be outlined from start to finish, with deviations from this plan discouraged (Hirsch 2005). To achieve this, early in the project a plan is developed that outlines the project strategy. In establishing desired artifacts from the engineering phase, the initial requirements and outputs are described in detail and included in documentation, such as contracts or written work agreements. In the design phase of the project, all design specifications are agreed upon prior to implementation with considerations for how this design will incorporate additional extensions. In the implementation phase, programmers follow the design plan, building software using plans developed in previous periods. After the software implementation phase is completed, the software is then tested to assess whether it meets the design specifications outlined in the beginning of the project. When the software implementation is understood to have met the requirements outlined in the plan and gone through a process of quality assurance, the software is then released to users. Developers then follow an established plan for maintaining this software for a set length of time (Hirsh 2005; Petersen and Wohlin 2010).

The plan-driven approach is typified by its linear structure, multiple phases where product quality is assessed, and an established plan that is carried through to the end of
the project (Hirsch 2005). If software changes are requested by either specific users or the entire user-base, the plan-driven process may need to be initiated again, starting from the planning phase of the project and ending in a new software release. Criticism of the plan-driven approach to software development asserts that the length of the new development cycle may not necessarily be responsive to users’ needs on appropriate time scales or be able to adequately adjust to specific user needs (Hirsch 2005; Petersen and Wohlin 2010).

2.2.2 User-centered agile software development

Agile software design (ASD) and user centered design (UCD) are two approaches to software development that were created from different motivations but seek to achieve similar goals: meet user needs in a timely fashion while also having the capacity to quickly respond to changing user preferences (Silva et al. 2011; Brhel et al. 2015). Plan-driven software development, described previously, was identified in some applications as too slow to respond to changing user needs; ASD was therefore designed to hasten responsiveness within an iterative software development framework by utilizing a series of quick feedback cycles (Petersen and Wohlin 2010). UCD, on the other hand, is less focused on development responsiveness speed, and instead emphasizes creating software that is usable by end-user audiences, another identified insufficiency of more conventional plan-driven software approaches. UCD relies on meeting user or customer needs, and therefore may require considerable time and resources to be invested in product research before the development period begins (Silva et al. 2011).

Integrating UCD and ASD attempts to address the deficiencies associated with plan-driven software development by playing to the strengths of both approaches. The results of these efforts to combine ASD and UCD is labeled as User-Centered Agile
Software Development (UCASD). UCASD is a method that is proposed to be both faster and better at meeting specific user needs than plan-driven approaches (Brhel et al. 2015). Silva et al. (2011) provides an example of how such an integration of ASD and UCD may unfold in a development process using the UCASD approach (Figure 2).

Figure 2: Integrating Agile Methods and Human Centered Design. Adapted from Silva et al. (2011).

In their review of the literature, Brhel et al. (2015) identify two central dimensions of UCASD, *process principles* and *practice principles*. Process principles include up-front analysis, such as planning and preparation phases, iterative and incremental approaches to development, and parallel creation tracks, where important features are addressed first, and lower importance features are addressed in future designs, either in subsequent development periods or in parallel to current development periods. Of note for study designs that promote iterative and incremental approaches to
software development, generating cycles of feedback where reactions from users collected in prior iterations is applied toward future iterations is identified as crucial (Sohaib and Khan 2010; Silva et al. 2011). To achieve effectiveness in iterative engagements, UCASD efforts are recommended to have short design cycles that represent incremental development improvements (Brhel et al. 2015).

Practice principles associated with successful UCASD efforts are observed to involve continuous stakeholder involvement and artifact-mediated communication. For effective stakeholder involvement in UCASD, stakeholder interactions are elicited early in the development process and remain active in their involvement in the development process in order to provide continuous feedback through the life of the project (Harris 2009). Recommendations on the degree and extent of stakeholder involvement, however, is a largely unexplored topic in the literature (Brhel et al. 2015). Artifact-mediated communication refers to the creation and distribution of comprehensive documentation, such as design concepts, prototypes, and other concrete and tangible outputs throughout the project so team members can be kept up-to-date about the development progress to facilitate interactions and communicate effectively with the project team (Nerur, Mahapatra, and Mangalaraj 2005; Kuusinen, Mikkonen, and Pakarinen 2012). Artifacts are seen as central to the software development process and become more salient when stakeholders from diverse backgrounds participate in UCASD to produce these artifacts (Brhel et al. 2015).

Challenges associated with the UCASD approach are numerous, with many of them revolving around time scarcity, difficulties in prioritizing development tasks, modulation of design tasks into achievable and appropriate time increments, and
management of work dynamics between developers and users/practitioners (Salah, Paige, and Cairns 2015). While many of these challenges can be mitigated through design considerations, suitable resource allocation, and guided interactions between developers and users, most organizations that undertake UCASD are time constrained, with developers, or a single developer, being shared across multiple projects and under considerable pressure and constant deadlines (Federoff and Courage 2009; Salah et al. 2015).

2.2.3 Design Frameworks for Visualization Software

While the above approach emphasizes iterative, user-interaction for software development that is generalizable to many software development tasks, such an approach is not necessarily optimal for meeting more specific needs, such as those identified in domain visualization. The nested blocks and guidelines model (NBGM) proposed by Munzer (2009; 2014; Meyer et al. 2015) is a framework for software development that specifically addresses domain-related data visualization needs. The nested model is a template design applied in four hierarchical levels, situated in different stages of the visualization design and evaluation process. These four levels include domain problem characterization, data/task abstraction design, encoding/interaction technique design, and algorithm design (Munzer 2009). Furthermore, NBGM is structured so the validity of the approach taken at each design level can be assessed (Munzer 2009; Munzer 2014). A graphical representation of these design components with examples of potential threats to validity at each of these design levels is displayed in Figure 3.

Each of the levels are described as having connectivity with flows from one level to the next, with potential interactions and feedbacks between levels (Munzer 2009;
Meyer et al. 2015). The first, or outermost level, is the characterization of the domain problem. The visual designer(s) identifies and targets a domain through their interaction with potential users, and in this process, develops an understanding of the domain as it pertains to the visualization problem. During this process of characterizing the domain problem, the designer gains competence in domain-specific vocabulary, the users’ existing workflows for handling data, and any user-related needs or requirements associated with the data to be visualized. For characterization of the domain, especially when engagement with users is paramount, in-person interviews and other data collection methods, such as user surveys and user observation, are utilized. The second level is the data/task abstraction design stage where domain problems, described in domain specific language, are mapped to generic descriptions and the vocabulary of computer science. This also includes the transformation of data types into data formats that can be easier to visualize. According to Munzer (2009) this abstraction stage is difficult to “get right” even when data/task abstractions are often carefully considered, with some designs operating under the assumption that “the first abstraction that comes to mind is the correct one” (p. 922) when many different abstractions may need to be tested before an appropriate abstraction is found. In the next stage, visual encoding and interaction design, the primary goal is to choose the appropriate design that communicates a suitable data/task abstraction to the user. As with previous design levels, for effective encoding/interaction technique design, feedback and interaction with the user group is essential. Finally, the last level, algorithm design, is when the visual encoding described in the previous step is written as an algorithm which automates the visualization process.
Munzer (2009) describes multiple threats to validity at each level of the nested model, as well as provides suggestions for how to address such threats. For the domain situation level, one threat of substantial consequence is the mischaracterization of the domain problem(s) articulated by users. This mischaracterization can be later reflected in low adoption rates by the intended audience, where tools are not adopted if they are not appropriate for addressing the specific domain needs of users. At the abstraction level, an example of a potential threat is that data transformation and processes generated by designers do not solve the domain problem of the user group. One way to combat such a threat is to have the user group test the software system using their own data and work flow, rather than rely on the designers to test its operation and assess its effectiveness. For threats to encoding and interaction, an example of a validity threat is that the design that was chosen does not properly communicate the abstraction from the previous level. Gathering insights into the effectiveness of abstraction communication can be achieved either through an informal process, such as communication and feedback with users, or more formally, through the implementation of software usability studies. Finally, threats to validity associated with the algorithm stage can stem from algorithm computational...
complexity, time performance to complete tasks, and memory usage, all which can result in a poor user experience and low software adoption (Munzer 2009).

While the nested blocks and guidelines model is a visualization design framework that involves iterative input and interactions from both users and designs in the visualization pipeline at multiple theorized levels, other approaches for designing visualization software include the formalization of a visualization design framework into an automated system, such as Draco (Mortiz et al. 2019). Draco is a system that considers a model for visualization using a machine learning algorithm specification that can be represented as a set of logical arguments formed as design constraints. These design constraints allow for the automation of visualization representations, including hard constraints (e.g., how data types are visually encoded) and soft constraints (e.g., time values that only appear on the x-axis and not the y-axis). Preferences of different visualization features can be learned by the visualization system, allowing for a ranking of visualization features for domain applications. By this system being continually applied, machine learning algorithms can then recommend visualization specifications that are domain specific, such a color profiles, range of displayed values, type of data representations, etc. A call for the integration of machine learning techniques to meet domain specific needs more generally has been proposed by other scholars (see KLAS; Peters et al. 2014; Peters et al. 2018).

2.3 Co-production of knowledge

This final section of the literature review describes co-production of knowledge, an approach, like the software development approaches described previously, that elicits perspectives from users in an iterative format. However, the way in which co-production
interactions between domain and non-domain participants are structured differs substantially from previously described software development approaches. In particular, there are differences in the way that problems and goals are identified and reevaluated throughout the co-production process. In this section, we first describe co-production of knowledge, its theoretical underpinnings, and its justification for applications in wicked problem contexts. We then review recommendations from recent scholarship on how to approach knowledge co-production activities.

In a normal science paradigm, communities that make decisions about the quality of science production are limited to those with similar disciplinary training. Science production is bounded by methods and approaches that are accepted by peer communities of discipline experts (Funtowicz and Razetz 1993). This normative approach views the goal of science production as an “extension of certified knowledge” (Merton 1942, p. 270) that is empirically confirmable and follows a methodology that reflects technical and moral norms about the common imperatives of modern science research. Increasingly, there are policy issues identified that involve risk or the environment that are both challenging to address and of high importance to society. The management of these crucial uncertainties are often incompatible with the methodological rationale associated with normal science (Funtowicz and Razetz 1993).

When uncertainty and decision stakes are high, the traditional methodologies for approaching problem solving that relies on the judgment of expert peer groups lose effectiveness. Single discipline approaches, by definition, are constrained in their scope of scientific knowledge, which may be insufficient for addressing certain types of problem contexts. In this post-normal problem context, development of extended peer
communities beyond discipline-trained peer groups is considered crucial to the production of scientific knowledge (Funtowicz and Ravetz 1993; Lemos and Morehouse 2005; Dilling and Lemos 2011; Meadow et al. 2015). Extended peer groups may include a group of interdisciplinary scientists as well as non-discipline trained experts that have local, experiential, or contextual knowledge. The overall intent of bringing together such experts from diverse backgrounds is to produce knowledge, an activity frequently referred to as knowledge co-production (Jasanoff 2004; Pohl et al. 2010; Armitage et al. 2011; Frantzeskaki and Kabisch 2016).

Another argument for the use of a post-normal approach is that a combination of discipline- and non-discipline trained experts with multiple perspectives can be more effective in confronting the complexity presented in certain problem contexts, including wicked problems. Wicked problems, originally characterized by Rittel and Webber (1973), are problems that are resistant to resolution, are unique (non-generalizable), and when a solution is found, this solution may uncover or generate other problems (Rittel and Webber 1973; Conklin 2006). A stated objective of finding a solution for wicked problems may not necessarily be appropriate, and instead a more suitable approach could be applying interventions that can address problems as part of an evolving process (Head 2008). Conceptualizations of wicked problems are often characterized as having high-uncertainty and high-decision stakes, and, like the post-normal paradigm described previously, one proposed approach for addressing wicked problems is knowledge co-production.

Knowledge co-production in a post-normal paradigm can also be understood within the context of a research system referred to as Mode 2 knowledge. Mode 2
knowledge is research that is produced “in the context of application” (Hessles and van Lente 2008, p. 740) through the efforts of collaborations across multiple disciplines (Gibbons 1994). This is in contrast to Mode 1 knowledge, like the normal science paradigm, which is produced in a single discipline academic context with quality control conducted through peer review. While a post-normal approach is similar in scope to Mode 2 knowledge production, there are a few key differences. While the Mode 2 approach emphasizes transdisciplinary research and self-reflexivity, it does not require the same co-production involvement from stakeholders or extended peer groups (Hessles and van Lente 2008). The validity of Mode 2 knowledge production has been challenged namely for its novel types of quality control, lack of generality, and lack of coherence within the different modes of knowledge production (Albert 2003; Hemlin and Rasmuessen 2006; Hessles and van Lente 2008).

There have also been more direct critiques of the joint knowledge co-production process in a post-normal paradigm. While co-production efforts may have a clearly defined social benefit, their usefulness for science production outcomes remains largely unknown and unexplored (Voorberg, Bekkers, and Tummers 2014; Hegger and Dieperink 2015), with a wide range of practices and attitudes applied that make it challenging to assess the effectiveness of the approach (Thompson et al. 2017). A survey of Dutch researchers involved in various joint production efforts suggest that merits of co-production tend to outweigh its drawbacks (Hegger and Diesperink 2015). Merits are identified as a wider empirical knowledge base, increased social debate about analysis and methodology, more research and stakeholder reflexivity, and the creation of a community of researchers and practitioners. One potential pitfall was identified as less
opportunities for publication output for academic researchers (Hegger and Diesperink 2015). Other research suggests that co-production efforts between science researchers and a restricted group of policymakers could be more tension-laden than groups comprised of stakeholders with more varied backgrounds (Lovbrand 2011). Co-production with climate scientists and a relatively restricted group of European Union policymakers was problematic for addressing areas that did not have direct policy relevance. In this respect, the co-production effort closed down rather than opened up climate policy goals by reducing the opportunity for recruiting participants that could have offered more diverse perspectives (Lovbrand 2011).

2.3.1 Co-production for ecology and environmental science

In the field of climate science, as well as environmental management and governance, there has been increased interest in applying co-production approaches (Lemos and Morehouse 2005; Ziervogel, Archer van Garderen, and Price 2016; Wamsler 2017). A growing body of scholarship has suggested that successful co-production exercises often establish strong linkages between researchers and users of knowledge, with effective collaborations often a direct result of the level and quality of interactions between groups of participants (Kirchhoff et al. 2013; Howarth and Monasterolo 2017). Co-production in a climate science and/or an environmental context is usually understood as interactions of different stakeholders from different facets of the science-policy-society interface, typically brought together with a common purpose of co-generating knowledge and information to be applied to environmental issues (Lemos and Morehouse 2005; Djenontin and Meadow 2018).
In a review of co-production of knowledge in environmental sciences, Djenontin and Meadow (2018) identify common themes that emerged across nine co-production case studies. The following elements were shown to either contribute to, or interfere with, the co-production process: context, inputs, activities, outputs, and outcomes-impacts. For co-production context, institutional characteristics, such as agency and management structures, as well as organizational and logistical factors were considered critical. For example, co-production contexts with more flexible institutional arrangements, collaborative cultures, and closer participant physical proximity were identified as components that laid the groundwork for successful co-production (Castellano et al. 2013). For co-production inputs, proficiency and expertise for the co-production process, legitimacy and trust, and inclusivity were all identified, with individual proficiencies and early trust developed through information sharing understood as essential (Cash et al. 2002; Foley et al. 2017). For co-production activities, this includes both a setting-up and development and design component. For setting-up, it is necessary that all the appropriate expertise has been identified and is included in the co-production process, and, within the development and design phase, all participants must be involved in the conceptualization, definition of methods, and planning of the project (Podesta et al. 2013; Kirono et al. 2014). When implementing co-production approaches, participatory teams work together to produce knowledge; projects that use engagement activities to gauge usefulness saw success, especially when maintaining flexibility and creating an environment where all participants can contribute to the project. In the outcomes phase, the communication of the produced information to all stakeholders is observed as critical, especially in a case where the science needs to be translated or contextualized in way that information can be
disseminated to all stakeholders (Cvitanovic, McDonald, and Hobday 2016; Young et al. 2016). Finally, beneficial outcomes and impacts associated with the project are identified as well as any best practices that were uncovered, with a conception about how relevant the co-production process was to stakeholders. Conventional metrics for measuring outcomes in research projects, such as number of academic publications or adoption of a tool by a broader community, may not fully capture co-production outcomes (Jordan, Ballard, and Phillips 2012). This suggests that for certain co-production activities, it may be necessary for researchers to “acknowledg[e] the success of the process and social learning as outcomes rather than an actual impact of practical changes from knowledge co-production” (Djenontin and Meadow 2018, p. 894).

There are limited instances of a co-production of knowledge framework being applied to the development of computer software for addressing wicked-type environmental contexts. For example, a co-production process was used for a web application development project for land managers to better access and apply climate data in their job duties (Brown and Bachelet 2017; Zanocco et al. 2018). In this co-production effort, land managers’ perspectives were elicited first at the beginning of the project, prior to the prototype development phase, to assess needs and give feedback about the type and format of information to be presented in web tools (Brown and Bachelet 2017). In a second iteration of this co-development process, managers gave more specific feedback about user interface design using web tool blueprints and web interfaces that were modified by guiding managers through a semi-structured interview protocol and holding a series of meetings with manager-users (Zanocco et al. 2018). While this example of a co-production of knowledge framework applied to web-applications is a
novel undertaking, at least with respect to broader literature considering the science-policy-society interface of co-production, it shares similar features to software design frameworks in computer science that emphasize iterative and user driven domain-visualization software products.

In this literature review, I surveyed and synthesized scholarship related to the scientific domain of environmental science and ecology, software development and software visualization design, and knowledge co-production. The lessons from this literature review were then applied to my research and are described in the subsequent chapters of this study. In particular, this review of the literature informed the topical areas of inquiry selected for inclusion in a codebook that was used to analyze data collected in this research project. The development of this codebook and its connection with extant scholarship is described in further detail in the next chapter (Methods).
3 Methods

In this research, I follow a longitudinal framework and apply a pre-post design using case study methods to collect qualitative data (Jensen and Rodgers 2001). In this chapter, I first describe the theoretical and methodological justification for pursuing such a research approach, and how it is appropriate for assessing the effects of the intervention I am considering. I then describe the research design and timing of the intervention, including the methods by which participant and other project data was collected. Next, I describe the development of an analytical codebook and how this codebook is applied to the data collected in this project. Finally, I describe the approach I have taken to mitigate various threats to validity and address potential ethical concerns.

Prior to undertaking any research involving human subjects, the proposed research design for this case study was approved by Oregon State University’s Institutional Review Board. Approved aspects of this research design included the recruitment of study participants, semi-structured interview questionnaires, information and data protection procedures, and other components of the research protocol. The project investigators who were involved with collecting, analyzing, or otherwise interacting with participant data all completed research ethics and compliance training offered through the Collaborative Institutional Training Initiative program (CITI Program).
3.1 Research Design

3.1.1 Longitudinal case study designs

This research considers the progression of a single case of a software development project across time. Because change across time is an integral component of this research, data is collected across time in a longitudinal design. Prior scholarship has discussed the value of using longitudinal data collection for understanding how complex dynamics unfold across a case or cases of interest (Jensen and Rodgers 2001; Saldaña 2003; Carduff, Murray, and Kendall 2015). A subset of longitudinal case studies is the pre-post case study design, where research designs are organized around an intervention, which may include a particular program, policy, rule change, etc. In pre-post case study designs, data are collected in multiple periods, comprised of at least one pre-intervention period and one post-intervention period.

Analyzing pre/post data collected in these periods has the potential for uncovering instrumental mechanisms, although this may not necessarily be the goal, or even be possible, given the selected case or cases and nature of the research design, the type of collected data, and motivations for scholarly inquiry. One area of scholarship that frequently employs longitudinal qualitative designs is public health with these designs often being applied to measure phenomena such as the effectiveness of health-related interventions or the impacts of long-term strategies for treatment (e.g., Lawton et al. 2008; Boyd et al. 2009; Pinnock et al. 2011; Carduff et al. 2015). While time constructs in such studies vary from months to years, selection of time scales is recognized as contextual in nature and there is no consensus on what length of time is most appropriate for inquiry (Saldaña 2003). In this research, we follow this approach to time scale
selection and use the context of the research project to form the pre-intervention, intervention, and post-intervention periods described in the next section.

3.1.2 Pre-post case study design and the VISTAS project

This pre-post case study design is structured around a development iteration of a visualization software platform called VISTAS (Visualization of Terrestrial and Aquatic Systems). The software development period that forms the basis for this research is identified as the intervention, which is referred to throughout this research as either the development period or development phase. It is important to note that while this research describes a single development period, the VISTAS software in this study has undergone many iterations of development, and in this research I am referencing the most recent pre-post sequence of this development process lasting approximately two years, starting in September 2016 and ending in August, 2018 (see Table 2).

To provide the reader with a sense of how this development process unfolded, Table 1 provides a proposed timeline for benchmarks in the development process, created at the beginning of the development period.

Table 1: Proposed timeline in early 2017.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VISTAS software co-development</td>
<td>Vector flows, visual analytics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Participant interviews</td>
<td>Prior to development (N=12)</td>
<td>After development (N=??)</td>
<td></td>
</tr>
<tr>
<td>Participant observations</td>
<td>Attend weekly development meetings, record interactions, collect documentation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Due to personnel changes, shifts in project development goals, and the overall fluid nature of the co-production process, this timeline was updated throughout the project. The finalized timeline is presented in Table 2. Note that in Table 2 the changes in timeline content from Table 1 are bolded in blue to ease in comparison.

Table 2: Finalized timeline in mid-2018

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VISTAS</strong></td>
<td></td>
<td>Vector flows, <strong>visual and data</strong></td>
<td></td>
</tr>
<tr>
<td>software co-development</td>
<td></td>
<td>analytics</td>
<td></td>
</tr>
<tr>
<td><strong>Participant</strong></td>
<td>Prior to development (N=12)</td>
<td>After development (N=11)</td>
<td></td>
</tr>
<tr>
<td>interviews</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Participant</strong></td>
<td>Attend weekly development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>observations</td>
<td>meetings, record interactions,</td>
<td>collection</td>
<td></td>
</tr>
</tbody>
</table>

3.1.3 Group context

The VISTAS Development Team is comprised of a diverse group of researchers and practitioners, including computer scientists, software engineers, and social scientists. The computer scientists and software engineers on the Development Team are well-versed in producing software for commercial and academic use, with experience in administering and executing software development projects. Two of the senior computer scientists on the project have been involved in other efforts to co-develop software dating back more than 20 years. In addition to members of the Development Team with expertise in computing, this team also included researchers from the social sciences. There are two social scientists on this project, which, including myself, have experience
with knowledge co-production. Notably, the senior social scientist has worked extensively on co-production projects with applications to environmental and climate science domains.

The VISTAS Development Team collaborated with two groups of participants, which we have described below as Environmental Team 1 and Environmental Team 2, and interfaces, sometimes indirectly, with a group of environmental science and visualization researchers referred to as Domain Experts. Environmental Team 1 and 2 are comprised of researchers who work together on projects as a unit, while Domain Expert participants are not engaged in a collective project. At the start of this VISTAS project iteration, Environmental Team 1 and 2 had agreed to be collaborators on the VISTAS project as software co-developers, while participants from the Domain Expert group provided their perspectives on visualization within their domain but did not participate in the software co-production effort.

Environmental Team 1 works within a government institutional setting that includes a lead scientist who has expertise in ecology and biochemistry, with other team members who are environmental scientists, Geographic Information System (GIS) specialists, statistical modelers, and computer scientists. This team conducts a wide variety of research activities, with a focus on bio-hydrologic modeling. Environmental Team 1 frequently works with stakeholders and other governmental partners, engages with researchers from a wide range of backgrounds, and as part of its duties advises and makes recommendations for land use decision-making in various biophysical contexts. This team conducts modeling that produces terabytes of spatiotemporal data, with this data structured in a regular grid that conforms to the spatial extent and resolution of a
chosen Digital Terrain Model. The resolution of the temporal data is selected by the modeler and varies from sub-hourly to yearly intervals.

Environmental Team 2 works in an academic setting, and includes a diverse group of researchers, with expertise in fields such as coupled human-environmental modeling, coastal hazards, social science, and emergency management and planning. Team 2 engages in alternative futures planning with governmental and non-governmental stakeholders in coastal areas, and engages in co-production exercises to build alternative scenarios that forecast future outcomes from climate impacts and human directed policies. Similar to Environmental Team 1, Environmental Team 2 conducts modeling that produces terabytes of spatiotemporal data. The spatial resolution of data outputs is dependent on geographic boundaries defined in the modeling process, which varies from a regular grid to irregular polygons. The temporal resolution of this data is defined by timestep outputs, varying from sub-hourly to quarter-century intervals.

The Domain Expert group represents an assortment of domain specialists who work in the areas of environmental science, ecology, geography, land management, and social sciences. While individuals in this group did not participate in the co-development of VISTAS software, their insight and perspectives were informative for understanding the data visualization process. For example, they provided input about their use of visualization tools in their domain-specific research contexts, their assessment of the landscape of existing visualization tools, and recommendations for improving communication with domain expert and non-domain expert audiences using visualizations. Data used by these participants vary from survey and interview data
collected from human subjects, to ecosystem models ran at world-scale that produce terabytes of output data.

3.2 Data Collection

3.2.1 Recruitment and interview context

Interview participants were selected from members of the four groups described previously (VISTAS Development Team, Environmental Team 1, Environmental Team 2, and Domain Experts). All interview participants were contacted by email and asked if they would agree to be interviewed, and if so, a time and location was determined. Of those contacted, only one did not respond to repeated requests.

Prior to the scheduled interview, each interviewee was provided with a short overview of the VISTAS project, my role in the project, and a brief description of the subject matter that would be covered in the interview. Interviewees were informed that they could end the interview at any time or could choose not to answer certain questions, and that their responses would be kept confidential. Participants were given the opportunity to ask questions or voice concerns about the interview process, and these questions and concerns were addressed before the interview began. After the participant consented to be interviewed and recorded, the interview commenced. Interviews were semi-structured, following an interview protocol that was developed at the beginning of the project (Appendices 8.1-8.2). Since this was a semi-structured interview, some modifications were made to interview questions based on prior participant responses and, in some circumstances, to appropriately capture the participant’s role and involvement within the project (e.g., changing wording like “scientific expertise” to “developer expertise” in the context of the interview).
Pre-interviews were conducted prior to the start of the development period, and post-interviews were conducted after the development period had concluded. In the pre-development period, twelve participants were interviewed, while in the post-development period, eleven participants were interviewed, with five participants completing both pre- and post-period interviews. Interviews were conducted in a variety of contexts and settings, including my office at Oregon State University and other facilities at Oregon State University, such as conference rooms and on-campus communal spaces, as well as personal offices of participants in institutes outside of Oregon State University. I have categorized these interview contexts as In-person, institutional setting (Table 3). Additionally, some interviews were conducted in-person and at non-institutional settings, such as a cafe, which I have categorized as In-person, informal setting. Finally, some interviews were conducted remotely via web-based conferencing software such as GoToMeeting™ and Skype™ using the video, audio, and screen sharing features of these applications, which I refer to as Remote, video conference. The verbal interactions in the interviews were recorded electronically and the contents of the interview transcribed verbatim by playing back an audio recording of the interview. When appropriate, additional notes were made in the interview transcript concerning interactions that were not identifiable in the audio recording. For example, when displaying a demonstration of the VISTAS software, I made note of the screen display contents when the demonstrations were referenced in the interview. Additionally, I made notes of pauses, laughter, and non-verbal cues (such as head nodding) when relevant to the interview.
Table 3: Interview summary information

<table>
<thead>
<tr>
<th>Summary statistics</th>
<th>Pre-development period interviews</th>
<th>Post-development period interviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interview count</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>Average Length (minutes)</td>
<td>44</td>
<td>64</td>
</tr>
<tr>
<td>Shortest (minutes)</td>
<td>27</td>
<td>37</td>
</tr>
<tr>
<td>Longest (minutes)</td>
<td>65</td>
<td>123</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context-setting</th>
<th>Pre-development period interviews</th>
<th>Post-development period interviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-person, institutional setting</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>In-person, informal setting</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Remote, video conference</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Interviews lasted anywhere between a half hour to two hours in length. Interviews in the pre-period lasted an average of 44 minutes while interviews in the post-period averaged 64 minutes. A contributing factor to the increased length of the post-period interviews was that these interviews contained a 15 minute interactive software demonstration where the participant was guided through the new features of VISTAS implemented in the development period, including exercises with participant data that returned results from Principal Component Analysis and Linear Regression. Table 4 provides a summary of metrics for interviews that were conducted in the pre-development period and post-development period.
### Table 4: Interview schedule

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interviewee</td>
<td>Date</td>
</tr>
<tr>
<td>VISTAS Development Team (DT)</td>
<td>DT (A)</td>
<td>11/7/2016</td>
</tr>
<tr>
<td></td>
<td>DT (B)</td>
<td>11/18/2016</td>
</tr>
<tr>
<td></td>
<td>DT (C)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DT (D)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DT (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ET1 (A)</td>
<td>11/30/2016</td>
</tr>
<tr>
<td>Environmental Team 1 (ET1)</td>
<td>ET1 (B)</td>
<td>11/9/2016</td>
</tr>
<tr>
<td></td>
<td>ET1 (C)</td>
<td>12/7/2016</td>
</tr>
<tr>
<td></td>
<td>ET1 (D)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ET1 (E)</td>
<td></td>
</tr>
<tr>
<td>Environmental Team 2 (ET2)</td>
<td>ET2 (A)</td>
<td>11/30/2016</td>
</tr>
<tr>
<td></td>
<td>ET2 (B)</td>
<td>11/28/2016</td>
</tr>
<tr>
<td></td>
<td>ET2 (C)</td>
<td>11/23/2016</td>
</tr>
<tr>
<td></td>
<td>DE (A)</td>
<td>11/10/2016</td>
</tr>
<tr>
<td></td>
<td>DE (B)</td>
<td>12/9/2016</td>
</tr>
<tr>
<td></td>
<td>DE (C)</td>
<td>2/2/2017</td>
</tr>
<tr>
<td></td>
<td>DE (D)</td>
<td>3/16/2017</td>
</tr>
<tr>
<td></td>
<td>DE (E)</td>
<td></td>
</tr>
</tbody>
</table>

1 This was a two-session interview that took place on 7/29/2018 and 8/29/2018

### 3.2.2 Data collected from project materials and participant interactions

In addition to data collected through in-person interviews, I also collected data that was generated as part of the co-production process. Written communication between the Development Team, and the Development Team and collaborators, took the form of emails, agendas, and meeting minutes. The text of all VISTAS-related written documentation was archived, including the date which it was generated and/or distributed. When this written documentation was combined it resulted in a single
document that was 129 pages in length, containing a total of 121 emails and 28 agenda summaries or meeting minutes (which were combined in analysis). Additionally, during in-person meetings between developers and collaborators I took written notes which were then combined with other team members’ notes, with particular attention to action items, future objectives, and/or development goals discussed in the meeting. Like other forms of written communication, these meeting notes were archived, and totaled eight documents. Because visualization design was such a prominent focus of this project, images from statistical outputs, videos of landscapes with model outputs, and screenshots from software displays were frequently distributed via email or presented in group settings. I collected these visualizations and made a note of when these visualizations were discussed in the context of the project, some of which are displayed in the next chapter (Findings). In total, 27 screenshots and 15 video clips were collected. Finally, over the course of the VISTAS project there were various presentations and other academic outputs, such as posters and conference papers, that either directly represented the VISTAS project or that featured visualizations produced by VISTAS software. When possible, these materials were collected and stored. During the course of my involvement with this project, this included the production of six posters, three presentations (which I conducted), as well as one conference paper. All collected project data, including in-person interviews and the project-related materials described above, were applied in the following analysis.
3.3 Data analysis: qualitative coding

3.3.1 Codebook development

To analyze data collected in this study, a qualitative codebook was developed. The main themes that comprise the codebook have two sources that reflect the unique research context under which this research project originates. A first family of themes refers to the development context of the VISTAS project. This development context is informed by co-production of knowledge and describes an iterative, evolving interaction between a software Development Team and their collaborators. A second source of themes described in the codebook are an extension of themes identified in a previous iteration of the VISTAS project. These analytical themes are described in detail below.

The VISTAS project has been active since 2011 and there was an Oregon State University graduate student, Kirsten Winters, working with the team at an earlier stage of the project as part of her dissertation research. I therefore had the unique opportunity to utilize information from her existing research, specific to the VISTAS project, and apply it to this current scholarly inquiry. Winters (2015), describes a codebook using an open coding method to analyze data collected from transcripts of in-person, semi-structured interviews and group meetings, where developers and collaborators met to discuss current progress and next-steps in a prior iteration of the VISTAS software development project. At the conclusion of open coding, Winters (2015) established four primary codes: technical challenges, exploration, communication, and development. Technical challenges, exploration, and communication were all identified as thematically-related to data visualization, while the development codes were thematically associated to the process of creating visualization software.
At the time of conducting her research, Winters (2015) did not examine the co-production process in her scholarly inquiry, and instead focused her research on elements related to domain-specific scientific visualization design and applications. However, in the recent iteration of the VISTAS software development process that I am considering, co-production plays a prominent role in both the design of the research and cognizant interactions between members of the VISTAS project. In order to generate congruency between Winters (2015) and this current iteration of the VISTAS project, I therefore modified the codes proposed by Winters (2015) to include co-production, which, in my conceptualization of this research, is an extension of development. This research contends that co-production of knowledge can be considered as an approach to software development.

To operationalize elements of co-production I used theoretical guidance from existing co-production scholarship. A recent review of co-production literature by Djenontin and Meadow (2018) identified the following elements that influence, contribute to, or impede, successful co-production activities (Figure 4).
Following guidance from Djenontin and Meadow (2018) on their proposed framework, Activities and Outputs are placed within Process, resulting in four primary co-production themes: Context, Inputs, Process, and Outcomes-Impacts. This is described in Figure 4 where Activities and Outputs are merged into a single Process coding category, delineated by a dashed box.

Combining these two coding schema from Winters (2015) and Djenontin and Meadows (2018) results in a total of eight codes under two separate categorizations. Secondary codes were selected based on Winters (2015) and Djenontin and Meadow (2018) with some modifications. For example, I first coded interviews using primary coding schema within the web application Dedoose™, and then used these coding categories to further identify in situ secondary codes using guidance from Winters (2015) and Djenontin and Meadow (2018). The final analytical codebook is described in Table 5.
Table 5: Finalized codebook

<table>
<thead>
<tr>
<th>Code Group</th>
<th>Primary Code (from the literature)</th>
<th>Secondary Code (in situ)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Visualization</strong></td>
<td>Technical challenges</td>
<td>Challenges with data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Challenges with producing visualizations from data</td>
</tr>
<tr>
<td></td>
<td>Exploration</td>
<td>Understanding data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Validation and troubleshooting</td>
</tr>
<tr>
<td></td>
<td>Communication</td>
<td>Analytics</td>
</tr>
<tr>
<td><strong>Co-production</strong></td>
<td>Context</td>
<td>Institutional factors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logistical factors</td>
</tr>
<tr>
<td></td>
<td>Inputs</td>
<td>Proficiency and expertise for knowledge coproduction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Legitimacy and trust</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inclusivity</td>
</tr>
<tr>
<td></td>
<td>Process</td>
<td>Setting-up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Development and design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementation</td>
</tr>
<tr>
<td></td>
<td>Outcomes-impacts</td>
<td>Outputs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changes in practice</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Salience of knowledge co-production</td>
</tr>
</tbody>
</table>

3.3.2 **Overview of the codebook**

The first section of this codebook describes themes associated with visualization challenges, including the main themes of technical challenges, exploration, and communication. The second section of this codebook describes themes associated with the co-production process, including context, inputs, process, and outcomes-impacts. In this overview of the codebook, I describe the main themes of the final codebook and a description of its contents. This codebook was applied to data collected in the pre-period, development period, and post-period when appropriate for this analysis.
Visualization: Technical Challenges

The themes of technical challenges include secondary codes related to challenges associated with managing and applying data, and challenges associated with producing visualizations from data. Data challenges described by participants were related to data types, such as data with a time component or a spatial component, and the size and speed of this data. Other data challenges included data properties related to uncertainty and the ability to process this data. While these data challenges were not directly related to data visualization, they were part of a data process workflow that enabled visualization. Such challenges are consistent with managing big data generally (Lazer et al. 2014; Gandomi and Haider 2015; Agrawal and Choudhary 2016) and big data in environmental science and ecology specifically (Peters et al. 2014). An additional, secondary code was formed to describe the challenges associated with producing visualizations from data, a difficulty described for both general and domain-specific applications (Munzer 2014; Winters 2015; Winters et al. 2016). Many of these challenges were associated with a current ability to produce visualizations, such as the ease of creating visualizations and the speed at which software can produce visualizations. An example of data categorized under the code of visualization challenges includes participants describing not having enough time to learn how to use visualization software because of its complexity due to many features that were not applicable for their use case.

Visualization: Exploration

The theme of exploration was associated with secondary codes including understanding data, validation and troubleshooting, and analytics, which arose from content describing participants’ domain-specific exploration needs. Content categorized
as understanding data included descriptions by participants of using visualizations to gain intuition about data, such as understanding how soil moisture and snow pack changes by viewing graphics combined with landscape displays. Content coded as validation and troubleshooting were mentioned in the context of verifying model outputs, for example, using visualizations to understand if a model that simulates sunlight and shadows was producing output consistent with other information about landscape features. Finally, the code of data analytics applies to the discussion of statistics and visual and/or data analytics coupled with visualization to aid in data exploration, such as selecting an area of a visual landscape display that then produces another window containing statistics generated for that area of interest.

**Visualization: Communication**

Themes associated with visualization communication were sorted into secondary codes related to communication with scientists and communication with non-scientists, which described the two main audiences that project participants interacted with when explaining their scientific work. Using visualization for communicating complex information to environmental science and ecologist audiences is recognized as critical (Cushing et al. 2015; Winters et al. 2016). Consistent with these conclusions, interview content related to communicating with scientists via visualizations were prominent, for example, participants described how a scientist audience did not fully grasp a phenomenon being presented until they were presented with a video displaying its impact on the landscape across time. Outside of academic settings, participants described frequent presentations to non-scientists using visualizations. An example of content
coded for non-scientist communication includes descriptions about how a non-scientist audience reacted intensely and positively to presentations that contained video imagery.

Co-production: Context and Inputs

The primary theme of co-production context refers to the conditions that contributed to the co-production effort, including secondary themes of institutional factors and logistical factors, adapted from Djenontin and Meadow (2018). Data coded as institutional factors were related to institutional type (e.g., government, education, research, etc.) and affiliations, arrangements, and partnerships that spanned across institutions. Content coded as logistical factors were associated with characteristics such as proximity, including the physical distance between institutions, and more broadly the constraining or facilitating factors that impacted the quality of interactions between participants in the co-production process.

The primary theme of inputs is subdivided into secondary themes of proficiency and expertise of knowledge co-production and legitimacy, trust and inclusivity (Djenontin and Meadow 2018). While the theme of context relates to the conditions for co-production, the theme of inputs refers to the characteristics of project participants. Proficiency and expertise relate to attributes of individual experiences with the co-production process, as well as personal competencies required to effectively produce knowledge in the context of the project. An example of data coded as proficiency and expertise for co-production includes statements such as a participant having developed software through interacting with non-computer scientists for a large portion of their career. The code of legitimacy, trust, and inclusivity reference interactions between participants that facilitated the co-production process. An example of coded content
includes how a participant felt comfortable bringing up any concerns or issues about the project with the lead developer, and how they were confident that this would have led to a thoughtful discussion.

**Co-production process**

The theme of process is critical to co-production activities, and secondary codes related to this process include setting-up and development, and implementation and outputs (Djenontin and Meadow 2018). Codes related to setting-up and development were associated with the practice of creating software, including the type and frequency of interactions. The secondary code of implementation and outputs were related to how the resulting software product was distributed and perceived by the users of the product and included an assessment of how knowledge was produced and its effectiveness for the targeted population. An example of content coded as implementation and outputs included participants describing the components of new features that were now part of software platforms, and how these new features were applied by scientist collaborators.

**Co-production outcomes and impacts**

The theme of outcomes and impacts relate to two subcodes, changes in practice and salience of knowledge co-production (Djenontin and Meadow 2018). Content describing changes in practice relates to how participants approach their work, and how the nature of this work is impacted through participation with the co-production process. For example, content coded as change in practice describes how interdisciplinary collaborations compelled participants to explain their thought process and work within teams of experts with differing backgrounds. The secondary code of salience of
knowledge co-production refers to the relevance of the co-production process to the project use-case, for example, how this co-production process compared to other software development approaches that could have been applied instead. Another example of salience of knowledge co-production relates the process of co-production to knowledge generation more broadly.

3.3.3 Data Analysis Strategy

This analytical codebook was applied to data collected across the three study periods delineated in this study: pre-development period; development period; and post-development period. Sources of data in the pre-development and post-development periods are comprised of transcripts from in-person interviews and project materials, while data in the intervention period is primarily comprised of project materials such as emails, meeting minutes, visualizations, and screenshots from prototype software displays.

3.4 Validity and Ethical Issues

3.4.1 My embeddedness with the VISTAS project

In this field study, I occupied dual roles as both an active contributor as a member of the project team and as a social scientist examining the process of co-production. In this dual role, I was both an observer of project meetings and participant interactions while simultaneously providing input about the direction and development of the software. Separability between these dual roles was established at the beginning of the project and was enforced or relaxed dependent on circumstance.
For example, when providing input to project participants I was careful to never reveal information that was disclosed to me in confidence. Relevant insights gained from interviews that were applicable to the development process were only shared in a way that maintained the confidentiality of their source. Aspects of the social science inquiry were also separated from the wider VISTAS project. For example, the content of the semi-structured survey instrument was not revealed to team members except for the principal investigator who submitted it for Institutional Review Board approval.

Additionally, I limited discussion of social science findings related to the direction and overall impact of the project and effectiveness of the co-production effort overall. When I did discuss findings, I did so in a way that minimized their scope and was less likely to influence participant perspectives. In particular, I attempted to avoid biasing participants toward any conclusions about the impacts of co-production. Their perspectives about the project’s impact were an important part of assessing its effectiveness and I did not want them to interpret their experiences through my own perspectives. Many participants, including the Development Team, were uninformed about the social science inquiry component of the project because this was rarely discussed in group meetings. Instead, conversations about the social science component of the project were limited to the principal investigators.

Much of my contribution to the VISTAS development process in both meetings and interactions with participants was of a technical nature. While I am not a computer scientist, environmental scientists, ecologist, or programmer, I do have knowledge of computer programming, having taken coursework using the Python programming language with applications to geospatial analysis, and have experience conducting
statistical and visualization tasks using programs such as R and Python. I also have taken computer-based cartography coursework, which covered topics such as geospatial and information visualization using applications such as Adobe Illustrator and Esri’s ArcMap. Having this knowledge base allowed me to contribute to this co-production effort as someone who has experience with visualizing data using computer-based software and techniques. Because of this, my involvement in this software co-production activity was truly immersive. On one hand, I was deeply involved with both the development, testing, and presentation of the software. On the other hand, I was assessing the progress of the development process using social science techniques, including semi-structured interviews.

Since I was involved in both software development, and assessment of the development process, my exploration into this process is likely colored by my own personal biases toward the VISTAS software and the development of this software. While, to the best of my ability, I have attempted to separate these two aspects of my involvement, I do not believe that complete separability is possible, or even desirable, from a research perspective. The VISTAS project embraced a co-production approach, and I was a part of this co-production effort, as opposed to an outside observer looking in. Through my involvement with the VISTAS project, I have cultivated personal relationships, and some close friendships, with the participants of this project. Indeed, I have an interest in seeing the product of the VISTAS project succeed, both because I was a part of this process and because I have a very real conceptualization of the time and effort that went into its creation. However, I also think it is possible for me to critically assess how this process unfolded, and fulfill my role as a social science researcher to
collect and analyze data, and generate insights in an impartial manner. It is only through the rigorous assessment of co-production activities like the VISTAS project can new lessons can be learned, and, in the spirit of co-production, new knowledge can be produced that is both useable and useful.
4 Findings

Findings are organized to reflect the progression of the VISTAS project across the study period considered in this research. Accordingly, it is organized in three phases: a pre-development phase, a development phase, and a post-development phase. In the pre-development phase (or pre-period), I analyzed interview data collected prior to when the 2017-2018 VISTAS development period began. These interviews are sometimes referred to as pre-period interviews to reflect the time period in which they were conducted. The next section reports findings from the development phase (or development-period). In the development phase, I analyzed data generated during the development process, which includes notes and communications between collaborators, meeting minutes, and outputs from feature and product iterations. In the final section of the findings, the post-development period (or post-period), I analyzed interviews conducted after the development phase had concluded.

4.1 Pre-period

4.1.1 Visualization

In the pre-development phase, the primary data collected was through participant interviews from Environmental Team 1, Environmental Team 2, the Development Team, and Domain Experts. Each member of these groups is assigned a letter for identification throughout the findings section. Please reference the Methods section (Tables 3-4) for more information about the interviews.
4.1.1.1 Technical challenges

Challenges with data

In the context of their participation with the VISTAS project, Environmental Team 1 primarily engaged with data generated by models rather than observed data. While these models were often calibrated using observed data, data output were typically the result of ecological and environmental process simulations run at landscape scales. These model outputs were explicit in both space and time, leading to the creation of spatiotemporal data (i.e. output units were spatially oriented and structured using interval-based time). Modeled spatial data outputs were typically in the form of a regular grid, or raster (e.g., Esri ASCII grid), while some analysis tasks applied vector-based layers (e.g., Esri shapefile). Digital Terrain Models (DEMs) played an important role in understanding how processes on the landscape advanced across space, with many of Environmental Team 1’s modeling activities applying DEMs to simulate hydrologic flow cycles in a watershed of interest. These modeled datasets typically ranged between a few square kilometers to hundreds of kilometers, with varying resolution, from pixels of tens of meters to hundreds of meters. The temporal resolution of the data outputs were also variable, with the time interval at which data were outputted dependent on model parameters specified by the user (e.g., hourly, daily, weekly, yearly, etc.) and related to the particular ecological phenomenon that was being explored. Environmental Team 1 member C described the spatiotemporal complexity of their model outputs by providing an example of nutrient transport in a watershed:

One of the outputs of the watershed model process are spatial distribution of nutrients in the soil at multiple layers. You want to know the amount of nitrogen in the soil at multiple layers over a landscape. And a landscape has trees on half
of it, and agricultural area on half of it…and then show how that changes throughout time (Environmental Team 1 member C, 2016 interview).

Environmental Team 2, similar to Environmental Team 1, primarily generated outputs from modeled data with observed data applied for calibration purposes. Environmental Team 2’s data stream includes many iterations of simulated future scenarios generated by interacting a set of alternative policies with climate projections, producing a volume of information that was regularly described by team members as overwhelming. As one participant from this group noted, the stream of information from model outputs “is like a firehose of data,” so much so that “at some point you just run out of time cycles to check every single metric” (Environmental Team 2 member A, 2016 interview). A metric is a term that Environmental Team 2 uses for tracking a particular category of output, which may include measures such as number of structures flooded, beach width, population growth, etc. For this size of data, which some participants described in terms of big data (e.g., issues arising from high volume data), there were challenges specific to interpreting and validating metrics outputted by modeling activities. Such challenges were related to the complexity of these model outputs not being adequately represented using available techniques, including data table summaries and visualizations, making it difficult to interpret these metrics in Environmental Team 2’s area of research interest.

A principal investigator for Environmental Team 2 (member C) further expounded on the nature of this firehose and expressed concern about the sheer volume of data, providing an example of the scale and scope of data they typically worked with:

We make simulations over about 100km scale on the outer coast plus a fairly significantly sized estuary…plus simulations are 90 years long. For any given sea-level rise scenario and a climate change scenario, we know that there is a
significant amount of variability on what the wave action may be, what the
storminess scenarios may be...so we hope to eventually get to dozens of model
runs for the spatial domain. In [Previous Project Name], this was on the order of
about 100 simulations that we ultimately were at and we were keeping track of
over 100 metrics (Environmental Team 2 member C, 2016 interview).

Given the sheer volume of this data, researchers who were part of Environmental Team 2
were uncertain whether they had ever truly gleaned all of the important insights that were
buried within the universe of their model outputs. Typically, to make the data more
manageable for analysis and reporting purposes, datasets were first filtered, with only a
subset of metrics (e.g., 10 of total available 100) being tracked in outputs from model
scenarios.

However, one participant from Environmental Team 2 saw the data challenges on
their project as more than just an issue that was related to the size and/or scope of the
data, but also the context under which new data was created. For this participant, a
challenge was developing a better understanding of existing data, not necessarily
updating data outputs or producing streams of new data:

I do think that too much data can be a problem, but I don’t know that that’s really
the nature of it. I think what it really comes down to is developing an
understanding of the data. I think that sometimes our inclination is to not explore
the information we have currently or fully, and to go generate more data. I think
ultimately that can be valuable, but you miss a lot of learning opportunities
(Environmental Team 2 member B, 2016 interview).

The VISTAS Development Team, often tasked with taking data sources from
collaborators and getting it into a format that can then be applied in a computer graphics
setting, identified data challenges of a different type. Rather than having issues with
interpreting streams of data output, especially with respect to aforementioned size and
scope of data, these challenges were more associated with taking data from collaborators
and transforming these data into formats that were appropriate for visualization. A
member of the Development Team with extensive experience working with computer graphics and a wide range of data types described it this way:

So I deal with a lot of data but really almost every time we do something new it is dealing with the data and the graphics just comes from this backend. And I tell my grad students and they don’t believe it. But then on the first project that they get they spend the first two-thirds of it dealing with the data (Development Team member A, 2016 interview).

Another member of the VISTAS Development Team echoed this sentiment about the time investment required to comprehend, and format, data for eventual visualization applications. However, the participant also stressed the importance of understanding the uncertainty of the data, in particular, the origin of the data and how it was created, stating “if you don’t understand where the data is coming from and what the data source is saying about it, you could be showing something that’s not really true, or something that has not been vetted very well” (Development Team member B, 2016 interview).

Similar to discussions from the Development Team, the Domain Expert group expressed data challenges associated with data uncertainty that often manifested itself in diverse ways. Two of the Domain Experts (B & C) described challenges specific to applying georeferenced data, and how this uncertainty can compound and interact in ways that are difficult to characterize. Domain Expert C, who frequently works with geospatial data, described these issues in terms of the scale of the data, asserting that because they “us[e] national data, there is a lot of uncertainty at various stages” especially regarding georeferenced survey responses:

Ideally I would have survey data of individuals georeferenced to points of their household location. That is usually the best case scenario. So sometimes we have to georeference to polygons like census tracks, or block groups, or ZIP codes, or counties or cities. But then there is also another set of environmental data that is usually georeferenced as raster data…say a one kilometer grid or finer resolution (Domain Expert C, 2016 interview).
Summary of data challenges

Participants interviewed in the pre-period identified multiple challenges associated with how they apply data in their workflows. These challenges included the volume and scale of data, the processing of data, and managing data uncertainty. Environmental Team 1 and 2 frequently worked with environmental models that produced largescale datasets at temporal and spatial scales, resulting in a firehose of information that was difficult to manage due to its complexity. The volume of information was an impediment to datasets being applied in further analysis that in some cases resulted in data being filtered as it moved through data analysis pipelines. The Development Team was mostly concerned with processing data for further use in visualization and issues with data uncertainty, also a challenge described by Domain Experts.

4.1.1.2 Challenges with producing visualizations from data

A lot of us don’t have time to learn Arc [Esri’s ArcMap]. I look over [Team member’s] shoulder and [Team member’s] shoulder and they have more buttons in that program than [ecohydrologic modeling software] does. That is saying quite a lot (Environmental Team 1 member A, 2016 interview).

The sentiment expressed by the above participant, who is the lead scientist of Environmental Team 1, is an observation echoed by other members of their team. Their team, which frequently interacts with large volumes of spatiotemporal model outputs, all expressed—to some degree—the technical challenges associated with visualizing their data using existing software platforms. Other members on Environment Team 1, with more expertise in using software programs like Esri’s ArcMap for the visualization of
geospatial data, still described the challenges associated with this process, especially for displaying topographic and time-series data.

Some challenges specific to topographic visualizations involved how to simulate three-dimensional space on a two-dimensional display while also representing relevant information draped on this topographical landscape. One way to produce such topographical representations is to use hillshading, a technique that simulates a light source that casts a shadow on landscape features. However, as one participant from Environmental Team 1 noted, hillshading techniques also has its drawbacks, stating:

Hillshade interacts with colors. [Team member’s] tools outputs colors, between white and black, say, that’s like a percentage of shade produced at the surface, right? So then if you add hillshade effects to make it 3D it actually distorts his data (Environmental Team 1 member C, 2016 interview).

Under this context, visualizing in 3D in a tool like VISTAS, which utilizes shaders, may introduce new interpretation challenges related to how data is visually represented on topographic displays.

Challenges with displaying temporal data were further described by a researcher on Environmental Team 1, and these challenges appeared to also conflict with displaying data using topographic representations. Prior to using VISTAS, a participant from Environmental Team 1 described how they had to write a program capable of manually outputting PNGs (Portable Network Graphics, a type of image file) from their model runs, and then “use other software to stitch those PNGs into a movie. It was more of a manual task…and it took that 3rd dimension away” (Environmental Team member B, 2016 interview).

Environmental 2 team members also expressed frustration with their current ability to produce visualizations using available software. Some of this may be due to
insufficiency in the functionality of existing software products, or at least for the software products that their team applied for visualization purposes. This could be due to the computational requirements of the domain, the scientist’s training, or a need for software products that incorporated analytics alongside map visualizations, described below.

For participants on Environmental Team 2, the computational speed that visualizations could be produced and displayed was a commonly identified challenge. This differed from the perspectives of Environmental Team 1 who were more focused on the speed that the user could generate visualizations through software interfaces (i.e. navigating available features) although both teams discussed software computational speed in producing visualizations on displays. One of the project leads from Environmental Team 2 specifically recalled the difficulty of using Esri’s ArcMap software to visualize model outputs on-the-fly for an area of interest that was flooded in a scenario simulation:

You want to zoom into where that first point, where that weak point was in the coastal defenses. And you draw a box in ArcGIS and zoom in and then the whole group has to sit there and wait ten seconds or so. And if you don’t draw the box correctly, it crashes...me as the PI [principal investigator] but not the one who is running the model it is difficult to see enough model results in enough places to get a handle on what’s going on (Environmental Team 2 member C, 2016 interview).

These issues with speed were discussed more generally as a limitation of the model data production pipeline, with visualization being just one part of this overall challenge. This particular challenge intersected with other thematic categories associated with visualization (communication, exploration), although these criticisms were often focused on the use and functionality of Esri’s ArcMap, a commonly used GIS product, within this pipeline:
We are at this stage where the researchers here at OSU are looking at outputs from the models. The main hurdle or difficulty is that it is a GIS format and so when you actually redraw the coastal flooding fields it takes a lot of time. So the simulations themselves might take a couple hours, but then to actually visualize the results of the flooding model, for example, where within the DEM [Digital Elevation Model] things actually flood it actually takes on the order of a few seconds to the better part of minute sometimes to actually have that pop up on screen…It’s not quick enough is one of the main issues. Part of it is it’s a GIS. It’s live drawing (Environmental Team 2 member C, 2016 interview).

Domain Experts also described difficulty with generating visualizations appropriate to their domain needs, with three of four participants mentioning that they needed additional personnel, often with visualization or computer science expertise, to generate a desired visualization(s). For example, Domain Expert C, who has extensive experience in visualizing geospatial data, does not have the technical skillset to create visualizations relevant to their science because it “require[s] some very specialized tools. It for one requires a programmer that is also familiar with spatial data” (Domain Expert C, 2016 interview). Domain Expert A, with experience with large scale ecological modeling, described an instance where to create 3D visualizations for an audience of land manager users required two programmers to build a 3D landscape simulator prototype. Working in a similar domain to Domain Expert A, Domain Expert C, who typically uses Microsoft Excel to create data visualizations, must utilize personnel within their institution to generate more complex landscape visualizations such as thematic maps, and defers to their expertise when designing such visualizations, stating that “Our GIS folks are pretty tuned in to what looks good and what doesn’t…” when producing map products and other data visualizations.

*Summary of challenges with producing visualizations*
Challenges in producing visualizations were frequently associated with inadequacy, or inefficiency, of available software to produce visualizations that participants identified as being important for their science. Commonly identified challenges included available and/or existing software not capable of adequately visualizing data on multiple scales, such as temporally and spatially. Additionally, software for producing visualizations were described as being computationally slow or overly cumbersome for the user to operate, with some domain-specific visualizations requiring specialists to generate.

4.1.2 Exploration

Data exploration was an integral component of research pipelines described by both Environmental Team 1 and 2. Since both teams utilized modeled data outputs, data exploration activities frequently involved calibrating model inputs, troubleshooting inconsistent model returns, and verifying model outputs. Both teams used visualization techniques for data exploration, however, at the time of pre-period interviews some participants from Environmental Team 1 had experience using VISTAS software (two of the three interviewed team members), while none of the three interviewees from Environmental Team 2 had experience using VISTAS software. Therefore, some of these interviews include participants who have had experience using VISTAS and viewing their data using the spatiotemporal features available in VISTAS, while others had not. Since the topic of conversation often was centered around the development goals for the VISTAS software platform, which utilized features for displaying topographic data in 3D as well as temporal data, many of the discussions with participants tended in these directions.
4.1.2.1 Understanding Data

For example, if we want to understand how soil moisture is changing, how snow pack is changing with different climate simulations that we might do, just looking at numbers in a chart would not be real helpful. So to screen data we use VISTAS…So VISTAS really offers us a way to better understand the data, in a first cut kind of way. So we load spatial output into VISTAS, we make movies, we use some of the new features to combine graphs with the spatial display. That has proven really useful, we gain a lot of insight in terms of how the environment is changing relative to human management of the landscape (Environmental Team 1 member A, 2016 interview).

The above quote from Environmental Team 1 member A, the lead of their team, is reflected by other members of the group, who often reported that simulation outputs were much easier or more intuitive to understand using VISTAS compared to other software visualization systems. VISTAS produced 3D displays resulting in insights such as “In 3D, it is like wow! I didn’t know there was that much harvesting going on…” When using other views for exploring data, such as a more common 2D display, complaints included “it is just like modern art in 2D” and that 2D relied too much on displaying visual information through “colors changing, it just doesn’t say much” (Environmental Team 1 member A, 2016 interview). When discussing exploring data outputs, another member of the team reflected that “rather than showing 2D plots of stream flow, you could show infiltration across the entire watershed” (Environmental Team member C, 2016 interview).

For Environmental Team 2, this exploration process first involved management and procurement of information in order to generate insights. For this team, even exploring simple modeling outputs can present challenges:

Maybe instead of looking at the results of three climate scenarios multiplied again by management scenarios and being overwhelmed by what the information would say about water demand, for example, may change over nine different scenarios. Or 27 different scenarios….we sort of played around with being able to turn
different datasets on and off with charts...that would be helpful when we think about these things in aid of exploration of the data (Environmental Team 2 member C, 2016 interview).

When asked if visualizations could help one of the modelers on Environmental Team 2 better understand their data, this participant did not see this as a valuable exploration tool, stating, “I don’t think it would help me. I don’t think personally it would help me understand the data. I’m really, intimately familiar with this data because I have to make it work” (member A, 2016 interview). However, this participant did say that visualizations would “help me communicate results to others” which in this case, involved enabling other members of Environmental Team 2 to better comprehend data outputs, which included data exploration tasks:

It would help them understand their data, help them explore the data, help them navigate through this huge amount of data...this huge amount of results, this huge amount of outputs. I think information and data is overwhelming when it is floodgated to people. I think [visualization] tools make it more accessible...not as helping me so much understand the data, but helping others consume the data (Environmental Team 2 member A, 2016 interview).

For developers, data exploration was a different venture than data exploration tasks among the scientists. For the VISTAS Development Team, data exploration was a process that occurred at multiple points of the data visualization pipeline, first as a way to gain intuition about data structure in order to put it in a format appropriate for display, and then to examine the data after it was visualized to understand whether the visualization was a representation of how they understood the data, and later, if this understanding was congruent with collaborators’ experiences. While this did not generate the same sort of insights as the environmental science teams since the goals for exploration were different, there was an overall interest described by a developer as “playing” in the data (Development Team member B, 2016 interview). While certainly
enthusiasm for varied and different types of data was of interest to the developers because it presented challenges and opportunities for visualization, there also was a sense that exploring the data itself was an exciting venture, with one participant stating, “I love to find people who have data. I’m a data junky and they are my suppliers, so to speak” (Development Team member B, 2016 interview). Another member of the Development Team (member A, 2016 interview) viewed visualization, and particularly the role that VISTAS has in the data science pipeline, as a tool for inquiry, applicable across multiple domains and to wide-ranging problem contexts.

Domain Experts, overall, discussed understanding spatial information, as many of the processes and phenomenon they were interested in evolved at spatial scales. For example, when trying to understand why a population of bird species was declining in an area, Domain Expert D “look[ed] at a whole variety of things spatially” (2016, interview). When discussing understanding spatial data at small scales, Domain Expert C, described how 3D was useful, but perhaps not crucial, to gleaning insights from data:

If you have the slope, you look at exposure…When I’m looking at the western U.S. it is important for me to see where the mountains are so having some sort of 3D is pretty important…but it is not essential…Instead of having to go and look at each point and seeing the elevation, we would know…it’s in a valley…or it is up high (Domain Expert C, 2016 interview).

However, for their work with spatial data that often did not involve topography, Domain Expert C did not see usefulness in 3D, but instead found that “people use a 3D visualization when a 2D visualization is more appropriate and easier to interpret, when they don’t actually have 3D data” (2016 interview). From this participant’s perspective, they frequently used visualizations like kernel density estimates or histograms to
“understand the distribution of [the] data” and for exploration of the data spatially they relied on 2D maps of point data (Domain Expert C, 2016 interview). Another participant, while using visualizations for communication and academic publications, “rarely used visualizations to understand data…when I think about visualization I think about how can I better present my results” (Domain Expert B, 2016 interview).

**Summary of Understanding Data**

The visualization of data was an important facet of participant workflows, with many of these participants describing how data visualization helped them identify and interpret phenomena contained within the data to apply in scientific activities. Some of the participants identified 3D displays as either helping them understand data or described the use of 3D in their current workflows. However, others, especially those who did not directly interact with topographic data in their research, saw less value in such information displays. Developers approached datasets in a different kind of way, with less emphasis on scientific understanding and more emphasis on understanding how data would be eventually applied by scientists in their workflows.

**4.1.2.3 Validating Model Results**

One user from Environmental Team 1 had extensive experience using multiple versions of the VISTAS software and served as an informal beta tester for the Development Team. A reason this user was so active as a VISTAS beta tester was because they were developing a sunlight irradiance model that was being coupled with an existing model in wider use within their research team. For the phenomenon this researcher was exploring, model outputs draped over terrain and displayed in 3D were critical due to the role of topographic features in producing shadows: “if you imagine that
you have a hillside and a tree on top of a hillside, depending on your solar angle then that
tree could cast a shadow all the way down that hillside to the base of what you are
looking at” (Environmental Team 1 member B, 2016 interview). They noted that
displaying model output in VISTAS became very important for validating phenomena
such as “Are shadows moving the right way, is the sun being modeled correctly?” since
topographic features could be used to corroborate whether this newly integrated model
was casting shadows appropriately and whether this was reflected in the integrated model
(Environmental Team member B, 2016 interview).

While some participants in Environmental Team 1 described the 3D display as
critical for validating their own model results, other members of this team saw the value
of 3D as “more useful as a communication tool. How I validate data is I usually have
field data that is collected at discrete locations, so I know what the runoff is…that is a
particular site location data, and I validate model results by pulling out model results
associated with that area” (Environmental Team 1 member C, 2016 interview). So, while
some members of Environmental Team 1 used 3D visualizations for model development
and validation, other researchers relied on different methods for completing these tasks.

Model validation for Environmental Team 2 had a different focus, often
comparing different model outputs against each other. In the case of Environmental Team
2, these comparisons may include different areas or biophysical contexts, making such
comparisons challenging from a model validation perspective. One participant from
Environmental Team 2 summarized this process as follows:

When you do that comparison in statistics, you are talking about measuring it in
two different ways and arriving at the same result, but not always. In this
particular instance, it is not just measuring it in two different ways, it is measuring
it in two different ways with different physical models and different
policies...How do you scientifically form a comparison? A scientifically valid, defensible comparison. I’m struggling with that now (Environmental Team 2 member A, 2016 interview).

Other members of Environmental Team 2 discussed validating models prior to showing “initial results to stakeholders…we are still making sure that the models make sense” that usually was a process of “trying to identify sensitivities of parameters…dense set of time evolution and spatial variability…in a comprehensive way” (Environmental Team 2 member C, 2016 interview). This type of model validation was important for the research group because when results were displayed to stakeholders there was often a second-tier of validation that required stakeholders to communicate knowledge and experiences based on presentations that included the visual display of landscapes and phenomena of interest.

For the Development Team, validation involved gaining an understanding of the conditions that lead to scientific conclusions, and whether this was consistent with expectations about model outputs. One of the developers discussed how they thought about the process of validation, usually through the eyes of their scientist collaborators, “In order to validate in [the scientist’s] mind…[the scientist] wants to know how that happened. What is going on around it, and what’s been going on around it for the last five or ten years…maybe there were things that were leading up to it” (Development Team member B, 2016 interview). This led a developer to suggest the use of a method of visualization called a *time extrusion*, discussed in the next section of the findings, that presents an alternative way of viewing timeseries data. Another developer discussed a similar dilemma that was identified by a member of Environmental Team 2, that to test the validity of model outputs, the user “can gauge the confidence of [their] models in
terms of whether [their] model agrees with someone else’s model” (Development Team member A, 2016 interview).

**Summary of validating model results**

Participants discussed validating model results using multiple methods, including using visualization systems to ground truth results and comparison of modeled data to observed data. Additionally, both environmental teams used visualization to compare outputs produced by different models to examine similar, or identical, biophysical contexts and processes. For developers, they relied on participants to validate that a visualization was depicting data in a way that reflected their understanding of model outputs or a research context.

### 4.1.3 Data Analytics

Incorporation of data analytics into the visualization pipeline was identified as a way that Environmental Team 1 could improve their ability to conduct data exploration. A desire of the team lead was to have a modular approach to data analytics functionality within a visualization system, with the capacity to add analytics features that met the demands of domain scientists:

So the important point is having the ability to be able to incorporate different kinds of analytic capabilities that are free and already pre-made. Maybe if there are novel ones, researchers can provide the ones that they build too. Be able to plug those in (Environmental Team 1 member A, 2016 interview).

While Environmental Team 1 generates data from models, rather than observed data, participants from this team saw the value of incorporating features capable of completing statistical techniques typically applied to observed data. From an analysis approach, Environmental Team 1 would “treat [ecohydrological model] output as real data, even
though it is simulated” and then apply this modeled data “in a way that statisticians might
to develop regression and predictive equations for estimating why something changed
given its position on the landscape or the vegetation cover” (Environmental Team 1
member A, 2016 interview). Additionally, the lead of Environmental Team 1 discussed
the potential of incorporating more sophisticated analytics related to habitat connectivity
into a visualization platform, stating how this would be applied for the “simulation of
landuse effects on habitat connectivity for whether it’s spotted owls or cougars” allowing
the user to “know the degree of connectivity from patch to patch as the landscape gets
broken up by development” (Environmental Team 1 member A, 2016 interview).

One solution proposed by a member of Environmental Team 2 was to create an
interactive visualization using a web-based application. While this was in the early stages
of development at the time of the interview, the idea behind such a web interface is that it
would allow team members, stakeholders, and other members of the attentive public to
quickly query model outputs in a way that helped distill complex temporal and spatial
features to make them more interpretable using data analytics. Of particular note is how
such an interactive web design would allow for a flexible presentation of data at different
scales and aggregation. One of the members of Environmental Team 2 described web
interfaces with integrated data analytics in this manner:

You click on a map, for a location, and it plots a chart. You might be able to click
a selection of the map, by highlighting a rectangle, and then it shows statistics on
that...All of these scales working together, and this map with a chart, with spatial
zooming, a point base, area statistics…you don’t have one form that your data is
presented in, you have many forms and they allow access to finer and coarser
spatial as well as finer and coarser time (Environmental Team 2 member A, 2016
interview).
While the use of 3D in the context of this project was typically associated with representing information draped over a terrain surface, where the third-dimension corresponded to elevation, other 3D visualizations that were not related to terrain were also considered. One of these non-terrain 3D visualizations was a temporal volume that within the VISTAS team was referred to as a *time extrusion*. In an earlier iteration of the VISTAS project, developers created time extrusions to help collaborators better understand trends that evolved across multiple timesteps. Instead of having to sequentially view changes by advancing through a series of single-period visualizations, a time extrusion uses one of the dimensions of a volume to represent time, allowing for a comparison of features as stacked slices within this volume.

Members of the Development Team also discussed the role of applying machine learning as an analytics tool to be able to help collaborators make sense of “so much data” and help scientists “know where the interesting stuff is” (Development Team member B, 2016 interview). In this developer’s estimation, one way to find “interesting stuff” or “unexpected stuff” is to “turn machine learning loose on it to decide what we don’t have to look at or where things are headed that appear that they may end up interesting” (Development Team member B, 2016 interview). When describing extending the capabilities of the current VISTAS software, another developer stated, “I think VISTAS itself could be a visualization tool that can begin to ask initial questions,” later stating how data analytics could allow the user to “ask more questions,” with a potential role for incorporating machine learning algorithms (Development Team member A, 2016 interview).
While seeing the value in integrating machine learning into data analytics, developers described more simple approaches, such as estimating “zonal statistics for an area” or use methods to “identify or filter out” different data classifications, types, and ranges of data values (Development Team member A, 2016 interview). This sentiment was echoed by a member of the Development Team stating that it would be “really cool if we had enough data mining expertise that we didn’t have to do it visually” (Development Team member B, 2016 interview). However, this developer conceded that among the tools and approaches that they were aware of, in the estimation of the developer, “a lot of times the best thing you can do with it is give scientists ways to look at it visually,” with the human power of observation “really good at spotting things visually in the domains that they know” (Development Team member B, 2016 interview).

**Summary of data analytics**

The team lead from Environmental Team 1 saw a potentially powerful role for integrating data analytics alongside visualizations, and suggested using analytics from statistical estimators. Environmental Team 2 had already began discussion about incorporating data analytics that they envisioned as web-based applications that team members, stakeholders, and the public could use to ask questions of the data. Developers, while acknowledging the value of less sophisticated data analytics, such as area or spatial summaries, described how applications of machine learning could advance collaborators’ ability to find interesting features in their data.

**4.1.3 Communication**

When discussing the effective communication of their scientific work, both Environmental Team 1 and 2 stressed the importance of using appropriate visualizations.
For these research teams, selecting visualizations was an exercise in understanding the intended audience, with scientific or domain-expert audiences requiring different visualizations than non-scientist or non-domain expert audiences.

4.1.3.1 Communication with scientists

When presenting work at academic conferences to an audience of scientists familiar with the domain, participants from Environmental Team 1 noted that certain visualizations, notably 3D visualizations that incorporated topography, were particularly impactful in explaining their models. One participant described the process of presenting their work to a domain audience of ecological and environmental modelers familiar with the physical processes that were being described in the presentation:

You talk about shadows and solar energy, people are exposed to it every single day but then when people think about modeling they don’t really think of what [light irradiance model name] or dynamic three- or two-dimensional modeling is doing. But then when you turn on a VISTAS video, people are like “whoa.” It’s just instant, they get it all of a sudden (Environmental Team 1 member B, 2016 interview).

This researcher also noted that other domain scientists in the field of hydrologic modeling were surprised by viewing such model results displayed in 3D. According to this researcher on Environmental Team 1 (member B, 2016 interview), this audience of modelers, familiar with water system modeling, “have a gridded network but they normally just have polylines loaded into Arc [Esri’s ArcMap] and they see where the stream is.” When this modeling audience saw spatiotemporal model outputs in a 3D visualization produced by VISTAS, they were “pretty taken back by it. And these are some advanced people who have made careers around modeling” (Environmental Team 1 member B, 2016 interview).
Environmental Team 2 was primarily concerned with communicating model results to a group of key stakeholders, and therefore were less focused on communicating their results to scientific audiences. However, one member of Environmental Team 2 drew a distinction between visual displays of information that were appropriate for interpretation in their scientific domain more generally, in this case, the use of timeseries plots (e.i. line plots with time on the x-axis and a metric of interest on the y-axis) for understanding model outputs, rather than spatial displays (e.g., aerial maps of landscapes). This observation about scientific interpretation versus non-scientific interpretation of visualizations was not explicit to communication challenges, and indeed touches on issues associated with data exploration:

Yes, I think from a researcher perspective timeseries plots are fairly straightforward…but that is probably not the best way for stakeholders to look at it. But for researchers, at least based on my background, that is how I want to think of the world. It is much more difficult for me to want to do something quantitative with map products. I get an intuition about the results but I gain a much more quantitative feel about how to handle the results and change the models based on timeseries results (Environmental Team 2 member B, 2016 interview).

While the developers did not discuss communication with scientists in detail, Development Team member B (2016 interview) did describe an instance when they were working with a group of medical professionals to improve imaging techniques. The developer had changed the visual presentation of information, by “color cod[ing] an x-ray as an experiment.” This change in the visual x-ray display evoked a strong response, where “the radiologist was just appalled…it was a real success, it looked great. It was a real failure because in the brain of the person looking at it meant nothing, it meant worse than nothing” (Development Team member B, 2016 interview).
When Domain Experts discussed visualization for scientist audiences, one of the participants stressed that it was important “to figure out what kind of crowd you are talking to” with visualizations being more effective within certain disciplines. In this participant’s view, people in “natural science engineering” when first reviewing an academic article “look at the pictures” and described how this may differ from people’s approaches from other scientific domains (Domain Expert B, 2016 interview). This same participant stressed that for the information presented in visualizations to be applied by other scientists, there had to be some vetting of presented information, such as a peer review process, to demonstrate it is built on “credible science” (Domain Expert B, 2016 interview).

Summary of communication with scientists

Members from Environmental Team 1 discussed how scientific audiences, when presented with new ways of visualizing model outputs, could still be surprised and gain new understandings, even if they were accustomed to working with such models in their own work. Environmental Team 2, and the developers, both considered how scientific audiences are accustomed to having information communicated to them in familiar ways, and deviations from the familiar presentation of information made interpreting this information more challenging. A Domain Expert made the distinction between different scientific disciplines, with these disciplines having different preferences for information delivery to their domain audiences.

4.1.3.2 Communication with non-scientists

When describing the value of communicating scientific work to non-scientist audiences using visualizations, a researcher on Environmental Team 1 emphasized the distinctions between communicating to different audiences, and the value of landscape-level visualizations more generally, stating “On one end you have researchers who are used to seeing tools like this. Not necessarily 3D but they are used to tools…but on the flip side you have a private landowner” (Environmental Team member B, 2016 interview).
interview). In this researcher’s estimation, “having them spatially visualize their landscape that they own and how it could be improved for the greater good” is a way of helping them develop a more holistic understanding of the watershed and how they are part of it (Environmental Team 1 member B, 2016 interview).

This same researcher on Environmental Team 1 (member B, 2016 interview) uses video of 3D landscape processes to communicate modeling results to non-scientist audience. When describing presenting results from a model to a non-scientists, the researcher describes a process where the video is shown once and the audience “qualitatively” connects to it. Then the researcher plays a video a second time, and there is a “shift to the quantitative, to understanding how these models are really explicit in both space and time” using a view displaying small scale granularity, such as a collection of cells. However, it was not until the third time that the video was shown did this researcher believe that the audience had fully accepted the model, stating “now that the model has been vetted and we have proved to them that they can use this as tool, then they are just immediately going to want to go back up and aggregate to a decision unit scale” (Environmental Team 1 member B, 2016 interview).

All participants from Environmental Team 1 were accustomed to presenting model results to stakeholders who did not necessarily have scientific training and were unanimous in their agreement that 3D visualizations provided a compelling way to communicate to this audience. Somewhat surprisingly, one member of Environmental Team 1, who had experience presenting results from VISTAS-produced videos, even commented that such videos were so effective that they considered this as potentially
problematic, putting a greater responsibility on the researcher to present accurate and high-quality data that was unambiguously communicated:

Every time they [non-scientists] watch a [ecohydrological] VISTAS movie, they are like “Wow, this is great!” Even if I know that…we forgot to fix this parameter or something, so I know this run is bogus. But still, people are compelled by it. So that is actually a dangerous thing too. I think that people are extremely compelled by pictures, spatial distributions and even more so by movies. Almost to a fault. They believe it. (Environmental Team 1 member C, 2016 interview).

Participants in Environmental Team 2 were also accustomed to presenting data to non-domain scientist stakeholders, often in public meetings that included members of both the research team and stakeholders. For a member of this team with extensive experience interfacing with communities, stakeholders, and non-scientist audiences, they found it hard to describe the visualization process more generally, but instead indicated a need for “realiz[ing] that there is no one single product. However visualized, however presented, however delivered, that is going to be the quote unquote right answer for a community. I think we need to look at it as a process” (Environmental Team 2 member B, 2016 interview). This Environmental Team 2 member further described this process of interfacing with stakeholders and “thinking about what questions are raised, having looked at that information.” This participant explained how this approach to communication and stakeholder interaction could be in opposition to the way that academic research is typically conducted, with academic researchers comprising the majority of Environmental Team 2. For this researcher, effective communication in co-production projects, and the type of projects that Environmental Team 2 engaged in also included the co-development of visualization products:

I think that’s the part where as a research community we really fall short, is that, seeing it as information and products delivered to a client in which case we want to close the book and move on. Funding is over, we have other grants and
proposals we need to be putting together, we need to move onto the next project. I think the real value is going through multiple iterations of the products. What maybe it is not about using a different chart or different visualization, it is using a different metric. Or using a different threshold. Or considering a different climate model. If we look at it as a learning process, the way the information is presented is a tool, the important piece is the process through this mutual learning or this coproduction or whatever you interpret it as (Environmental Team 2 member B, 2016 interview).

Another member of Environmental Team 2 discussed the challenges of communicating uncertainty and variability in the products that included the visual display of model outputs. For this researcher in Environmental Team 2, communicating this uncertainty and variability was critical for understanding model outputs because the future scenarios that were being depicted were probabilistic rather than deterministic, and this researcher did not want to misrepresent the team’s scientific confidence in projected results. However, rather than making the scientific distinction between uncertainty and variability, the approach that this team took was to describe both concepts as variability and display a range of values for metrics of interest:

We spent a lot of time trying to design our product such that they show uncertainty as well as we could and the way you do that is showing exactly the same map views, but from different scenarios, and so you start seeing the differences, that’s a variability. With these different scenarios you get different results…but in terms of maybe a more true uncertainty parameter, we have no idea what the climate is going to be so we have to run multiple simulations. So you have a best guess timeseries of a particular metric and then an error bar on that. We didn’t describe to stakeholders what that error actually meant…but we had said, this is the variability. For the most part people took that and at least didn’t complain that it is was overwhelming and too much information or too confusing (Environmental Team 2 member C, 2016 interview).

In some circumstances, the veracity of the information being presented was challenged directly by stakeholders, especially if information that was being presented differed from their knowledge and experience. In one such example, a member of Environmental Team 2 described an interaction with a stakeholder where the information
being communicated was misinterpreted, likely due to the way the information was communicated to the stakeholder group:

I was working with a fishing community and I showed a map where 80% of the fishing effort was made. Somebody looked at that and said that’s total garbage because I know for a fact that I fish in other places. And it is not shown here. What was misinterpreted is that there is 20% of fishing effort that we are not representing here, or the timeframe that the effort occurred on (Environmental Team member B, 2016 interview).

Development team members had less experience directly interfacing with non-scientists, as they were typically one level removed from scientists’ interactions with stakeholders. However, one Development Team member (member B, 2016 interview) was primarily concerned with the level of abstraction that a non-scientist audience could comprehend, and how this may vary from scientists who are more familiar with thinking about their model outputs, and visualizations of these model outputs. This developer discussed a member of Environmental Team 2 using a time extrusion visualization that displays temporal data as a volume that can be manipulated in three dimensions:

If you show one of these time extrusions to [lead of Environmental Team 2], [they will] say I see this and that, and this is happening here. I imagine if you show that to a member of the general public they will go ‘what the hell am I looking at. What do you mean that time goes up.’ I think we have to be careful about the level of abstraction…we can do a lot of fancy stuff…but I think there is a suspicion about science that if it looks too cool, too trendy, that they suspect you are just playing (Environmental Team 2 member B, 2016 interview).

Domain Expert C frequently communicated with non-scientist audiences and helped with implementation of web tools designed for general public use. The web tools displayed public opinion maps of the United States, with disaggregated spatial units that included levels such as state, county, and congressional district. The participant described how these visualizations were used for communication:
For the public, we are not expecting someone who is coming to see our maps to know what the national percent of people who think global warming is happening is. They just see a map showing major differences in colors…they might misinterpret the national level numbers, which is like 65%, if on the map they see a lot of blue and a lot of red…because the way we have this interactive map set up, you can click between questions fairly easily and having the same color scheme across all of them allows for comparing absolute, national-level differences…the downside of that is it is harder to see spatial variation within each question (Domain expert C, 2016 interview).

While Domain Expert C designed visualizations for general public consumption, Domain Expert D (2016 interview) described how the scientist vs. public distinction did not apply to their field, and instead, “the same visualization goes out to both,” with the audience for visualization products comprising “a combination of internal, so people working for [institution name], and external, the general public.” For this participant, many of the visualization design considerations revolved around the age of those consuming the visualization, with “younger people [looking at visualization] on computer screens” and “older people prefer[ring] to have paper” and how designing visualizations that could be displayed on screen displays, as well as in print in black and white, was a primary challenge.

Summary of communication with non-scientists

Environmental Team 1 discussed the effectiveness of using dynamic 3D visualizations for communicating to non-scientist audiences, however, members of this team also cautioned against presenting information that was not thoroughly vetted because such visualizations were so convincing that audiences may be too accepting of results even if they were inconsistent with their understanding of landscape processes. Environmental Team 2 described how communicating uncertainty to non-scientist audiences is challenging, and how visualizations may need additional explanation and
contextualization by scientists when presenting this information. A member of the Development Team described how abstract spatial visualizations, less grounded in recognizable characteristics, may prove difficult for non-scientists to interpret. One domain expert discussed selecting an appropriate visual (color) display for the general public, while another domain expert did not make distinctions between scientists and non-scientists and was more concerned with producing visualizations that could be both printed in black and white and viewed on digital displays.

4.1.4 Co-production

Since these findings pertain to pre-period interviews, before the development process had begun, content related to co-production were less frequent than visualization at this stage of the project. However, co-production themes that emerged in the pre-period were primarily related to inputs, described below.

4.1.4.1 Inputs

Proficiency and expertise for knowledge coproduction

For co-production inputs, members of Environmental Team 1 expressed confidence in the ability of the Development Team to be responsive and knowledgeable about the software production process, with one collaborator commenting on their previous interactions with the developers:

As a beta tester my job for the whole group was to use the software [VISTAS] normally but when things went wrong or I would get stuck… I would just write them [the developers] and [see if] they could get to the same issue. They addressed pretty much anything that I found (Environmental Team 1 member B, 2016 interview).
Another member of Environmental Team 1 also discussed the proficiency and availability of the VISTAS developers in addressing prior needs in earlier iterations of the project, describing how they envisioned a new feature being added to VISTAS software, stating:

Having a connection with the VISTAS group...and be able to say “Hey, we’d really love this [feature] to be linked”...just as we have been working with the VISTAS group all along or say “It would be great, [developer name] and [developer name] to have this or that.” They have been really responsive (Environmental Team 1 member A, 2016 interview).

In assessing the progression of their own skill as a developer, and their ability to contribute to the development process overall, a junior member of the development reflected that:

I would say that my skills in visualization have been exponential. I started with a little bit of nothing and now I know a lot more that I didn’t know. Especially in terms of 3D visualizations, but also the basics. Charts, graphics, how you implement the kinds of things and how you choose which colors to use (Development Team member A, 2016 interview).

Summary of co-production inputs

While infrequently discussed in the pre-period interviews, participants from Environmental Team 1 discussed their confidence in the technical proficiency and accessibility of the Development Team, referencing prior interactions where the developers were responsive to their questions and issues regarding the use and implementation of new VISTAS features. Additionally, one of the developers discussed how their ability to generate visualizations had much improved since the start of the project, an indication that competence in visualization and visualization software development had improved and will likely continued to improve with their continuing involvement in the VISTAS project.
4.2 Development period

During the development period, the Development Team interfaced directly with Environmental Team 1. While the intention in the pre-period was to collaborate with both Environmental Team 1 and Environmental Team 2, Environmental Team 2 did not participate in the co-production effort, except for a few interactions with one of the VISTAS developers to update an existing feature of VISTAS that facilitated importing data formats, specific to model outputs produced by Environmental Team 2, in the VISTAS software. While interviews were not conducted in the development period, interactions between the Development Team, and Environmental Team 1, were captured in meeting minutes, email communications, and other notes and memos taken contemporaneously (see Methods for more description of this content).

The focus of the development period was co-production, and, in particular, the setting-up, development and design, and implementation of knowledge produced in this time period in the form of new VISTAS software features. Because of this, there are no visualization themes coded in the development period, and the focus is on the co-production process. The outputs from this development period will be discussed in the post-period section, which follows this section, where post-interviews were conducted with participants.

I was involved at every stage of the development process, through my attendance in weekly development meetings and my interactions with participants and developers. Additionally, I influenced the development and design of VISTAS software, providing input on the conceptualization of features, interpretation of feature output, and visualization design. My dual role as a contributor to this development process and social
scientist conducting research into this process, and how I addressed potential sources of bias and threats to validity in my research, are discussed in Section 3.4.1 of the Methods chapter. To help orient the reader to the events that unfolded during the development period, a brief project history of this period is first described. Following this project history, the process of co-development during the development period is reported by analyzing this process using co-production themes, which includes my role as a member of the Development Team.

4.2.1 A brief history of the development period

4.2.1.1 Starting out

The development period described in this study began in December 2017 and ended in June 2018. This approximately half-year period represented a substantial effort by both the Development Team, and Environmental Team 1, to create new features to be implemented in the VISTAS software platform. The beginning of this period is also marked by a new software engineer joining the project who took on a majority of software development tasks. Before engaging the scientist team, the Development Team held multiple meetings to discuss the content of these mediated interactions, which occupied December, 2017 and early January, 2018. During this time period the new Development Team member was brought up to speed on both overall project objectives and software.

4.2.1.2 Development begins

Starting in mid-January 2018, the Development Team began iterating with Environmental Team 1 through in-person meetings and email conversations. Through
these interactions, a clear direction forward was established: the Development Team would work on data analysis feature development that included linear regression and Principal Component Analysis (Meeting Notes, 12/18/2018). During February and March, the developers received feedback from Environment Team 1 on various mock-up visualizations using existing statistical packages available in R and Python. Using this feedback, the developers created a working prototype that incorporated the new data analysis features by the end of March 2018.

4.2.1.3 Development continues

In mid-April 2018, the Development Team presented Environmental Team 1 with a working VISTAS prototype with new data analytics features, and received feedback based on its features and functionality. This feedback included changes to the way the scale of the information was presented, issues related to presentation of results, such as axis labeling, and extended the analysis to accounted for land use types in the form of categorical data (Meeting Notes, 4/16/2018). New features involving scale and presentation of data were developed and added to VISTAS, but incorporation of categorical data as a feature of the analysis was not completed due to multiple reasons, including ambiguity about preparing data for such an analysis and results reporting that required non-generalizable interpretation.

4.2.1.4 Development reaches an endpoint

The newest version of VISTAS was released and made publicly available for download in early May, giving Environmental Team 1 approximately one month to explore the latest VISTAS version with data analytics features and incorporate it in their
workflows. The final group meeting of this development period was held in June 2018 where the developers and Environmental Team 1 discussed issues related to the current state of the software, such as identification of software bugs, and possible extensions of existing features to include additional visual and analytical functionality. In the final minutes of this meeting, next steps forward were discussed including the future of the project and potential ways to ensure longevity of the project and software under upcoming budget constraints (Meeting Notes, 6/20/2018).

At this development endpoint, much of the available funding for VISTAS software development was depleted, with a small portion of the budget kept in reserve to address bug fixes, documentation needs, or any other unexpected issues that may arise.

**Summary of a brief history of development period**

The development period began in December 2018 with the addition of a new Development Team member. Based on feedback from collaborators, the Development Team implemented two new data analytics features in the VISTAS software: Linear Regression and Principal Component Analysis. The newest version of VISTAS, with added features, was made available in May 2018 and in the final meeting of the development period the group discussed bug fixes, possible extensions, and future project plans under a constrained budget (Meeting Notes, 6/20/2018).

4.2.2 Co-production

Below is an analysis of the co-production process that occurred during the development period using themes from (Djenontin and Meadow 2018). All of the co-production themes fall under the primary theme of process, since this corresponds to development-period activities, including setting-up, development and design, and
implementation. Outcome, while a secondary theme of process alongside setting-up, development and design, and implementation, is discussed in the post-period section of the findings.

4.2.2.1 Setting-up

At the time of the development period considered in this research, migration of the VISTAS software language from C++ to Python had already occurred. Python is a programming language with many scientific libraries available for both data analysis (e.g., SciPy) and visualization (e.g., Matplotlib), and, at the time of language migration, was chosen due to its perceived advantages for implementing data and visual analytics features in future development. After this effort was completed in August 2017, the discussion in late 2017 revolved around “integration of data exploration and/or statistic tools” (Meeting Notes, 12/13/2017). In development meetings, we considered designing a feature that would allow users to provide their own script, with options to take input data from VISTAS-produced visualization output and “produce a return value, such as new data” and “produce a graph” (Meeting Notes, 12/13/2018). Additional features discussed in late 2017 development meetings included “removing bugs”, “making UI (user-interface) easier and more intuitive”, “documentation”, and “raster calculator” (Email Communication, 12/18/2017). During our interactions with Environmental Team 1, we were primarily collecting input about scripts/features and other features that this team would like to see implemented in the VISTAS software platform. This would be the topic of multiple development meetings, and interactions between developers and Environmental Team 1, in the time period from January through April 2018.
Prior to the start of the development period considered in this research (December 2017), the VISTAS team was searching for a new software engineer to work on the VISTAS project, as a junior developer, who had been activity involved in VISTAS software development in 2016 and 2017, had found new employment and was no longer available to work on the VISTAS project. The junior developer who was being replaced on the project had gained substantial expertise in geospatial visualization, both in their work on the VISTAS project and developing web-applications that displayed 3D environmental information for an unrelated project through their work at a research institute. As a replacement for this software engineer, we recruited another software engineer from the same research institute. This software engineer was aware of VISTAS software through interactions with VISTAS Development Team members at their research institute and had extensive experience with software development involving environmental data, but less experience with geospatial visualizations.

Both prior to and during the initial stages of the development period, Environmental Team 1 expressed interest in hotspot detection, which, to these scientists, was a way to detect characteristics or areas on the landscape associated with high (or low) impact of a phenomenon. Conversations with Environmental Team 1 in January and early February 2018 included discussions around using machine learning approaches, and other techniques, such as implementation of Getis Ord Gi, a spatial statistical analysis method available in Esri’s ArcGIS software suite, to detect hotspots on the landscape. These efforts around hotspot analysis were ultimately abandoned in favor of feature design that the Development Team, and Environmental Team 1, thought achievable given the available budget that covered approximately two months of full-time development on
the project. Ultimately, through in-person meetings and multiple email communications between the developers and Environmental Team 1, development of two new statistics-based features emerged by mid-February 2018. These features included methods for conducting multiple linear regression and Principal Component Analysis (PCA) while concurrently displaying the landscape data that was being visualized.

**Summary of setting-up**

As part of setting-up for software development, multiple discussions revolved around feature additions and new functionality for the VISTAS software. This included bug fixes, improvements to the user-interface, and a raster calculator, as well as methods for detecting hotspots. After a new software engineer was added to the project, a set of objectives were agreed upon that were within the budget and feasible with existing project personnel. These objectives were the development of linear regression and Principal Component Analysis features.

4.2.2.2 Development and design

As part of the development of statistical features for VISTAS software, we first underwent a discussion about the design of such features. Linear regression is a common statistical technique used across many disciplines; however, this particular implementation of linear regression was specific to environmental science applications. Compared to linear regression, PCA is a less commonly applied analysis technique and is varied in its use across scientific disciplines.

The first decision to be made about development of statistical features was the data source(s) to be analyzed. After discussion with Environmental Team 1, and the developers, one viable option emerged with possibilities for extensions into other data
formats, including analysis on filtered data, in later iterations of the project. Given that
data displayed on topographic landscapes in VISTAS is temporal, that displayed data
changes based on the timestep of the data; this was the source of data that the team
converged toward.

Linear regression

After further communications with Environmental Team 1 about linear
regression, it became clear that the team had not previously applied this technique to
outputs from their ecohydrological model. To facilitate discussion about linear regression
applications to their data, I used R to run analysis on a dataset provided by Environmental
Team 1. I have experience running regression analysis in R, as well as familiarity with
generating visualizations using R packages such as ggplot2 (Wickham 2009). Since
Environmental Team 1 did not have a recommendation for post-estimation regression
visualizations, we first needed to decide what visualizations would accompany the linear
regression feature. To get an idea about the visualizations that were provided in base R, a
widely used statistics and programming platform, a simple linear regression model using
Environmental Team 1’s data was fit and then plotted. This resulted in five plots: Normal
Q-Q; Residuals vs. Leverage; Residual vs. Fitted; Scatterplot with Best Fit line; and Scale
Location. The plots that were used as examples in the development process are displayed
in Figure 5 - Figure 9 and were distributed to Environmental Team 1 via email in mid-
February 2018 (Email Communications, 1/22/2018 – 1/28/2018).
Figure 5: Normal Q-Q plot

Figure 6: Residuals vs. leverage plot
Figure 7: Residuals vs. fitted plot

Figure 8: Scatter plot with best fit line
After these visualizations were presented to Environmental Team 1, and discussed internally among the Development Team, the consensus was to create a visualization similar to Figure 8: Scatterplot with Best Fit Line. Further refinement of this scatterplot in R resulted in the following visualization examples, where the alpha level, or transparency, of the points representing observations were altered. This became a topic of discussion because with some datasets, particularly those with a large number of observations, or observations with the same or similar values for the chosen independent and dependent variables, a stacking of points made it challenging to determine the number of observations corresponding to x- and y-axis values. See Figure 10 and Figure 11 for an example of the same data and plot parameters with different alpha values for points.
Figure 10: Scatterplot with alpha = 1

Figure 11: Scatterplot with alpha = 0.1
In addition to visualizations of linear regression outputs, another consideration was the display of important outputs from linear regression models, including reported information on variable coefficients, standard errors, probability values, etc. To get an idea about the linear regression model outputs that are reported in post-estimation, we again turned to R, which provides a summary of linear regression models as a base function. We then provided the following output to Environmental Team 1 (Figure 12).

Upon seeing this regression output, one of the scientists interpreted the relationship between the modeled variables and commented that “it's questions like this that make this analysis so exciting. This is the first time regression analysis has been applied to [ecohydrological model] spatial-temporal outputs, so it will open up new possibilities for thinking about the results” (Environmental Science Team 1 member A, 2/12/2018, email communication).

Figure 12: Linear regression output from R

```
> summary(m1)

Call:
lm(formula = SpatialNitrif_t1 ~ SoilMoisture_t1, data = CB_2001)

Residuals:
     Min      1Q  Median      3Q     Max

Coefficients:         Estimate  Std. Error   t value  Pr(>|t|)
(Intercept)    1.542e-04   2.492e-05  6.189   7.48e-10 ***
SoilMoisture_t1 3.819e-04   4.033e-05  9.469     < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0002582 on 1776 degrees of freedom
(3984 observations deleted due to missingness)
Multiple R-squared:  0.04806,  Adjusted R-squared:  0.04752
F-statistic: 89.66 on 1 and 1776 DF,  p-value: < 2.2e-16
```

>
Based on the general format and content of these visualization and post-estimation regression outputs, the Development Team first created a VISTAS feature that was a simple linear regression (i.e. one dependent variable and one independent variable) and later added a feature that included multiple linear regression (i.e. one dependent variable and multiple independent variables). In the process of feature development, there was added functionality to have different scale options on which the data was explained, either as a fixed scale in which the y-axis values were set using two standard deviations of the minimum and maximum of the entire dataset, and an adaptive scale where minimum and maximum values for the y-axis changed based on the current timestep of the data that was displayed. Additionally, visualization functionality addressed the issue of very large datasets, and clustering of point data, by offering an option to display data as points in a scatterplot or as a heatmap with rasterized counts of data within cells of the visual display. The visualization feature for linear regression was available for simple linear regression when there were two variables being displayed (i.e. an independent and a dependent variable) but for more than two variables included in the regression there was no visualization display. See the figures below for an example of simple linear regression using two different datasets provided by the participant, with the best fit line displayed in red. Figure 13 displays a smaller dataset (n = 1778) where the scatterplot option is selected, while Figure 14 displays a larger dataset (n = 23,2763) where the heatmap option is selected.

Through discussions between the Development Team and participants, the following post-estimation regression output were identified as important for understanding the relationships of the modeled variables: variable name, the model
coefficient, standard error, t-statistic, probability value, and 95% confidence intervals associated with included independent variables, as well as the constant (or y-intercept value). Additionally, the number of observations, and R-squared, a measure of goodness-of-model fit, were included in output fields.

Figure 13: Linear regression scatterplot (left) of visualized landscape data (right)
Figure 14: Linear regression heatmap (left) of visualized landscape data (right)

**Principal Component Analysis**

Principal Component Analysis (PCA) underwent a similar feature development as described with linear regression, with linear regression and PCA development occurring concurrently. Some of the features for linear regression and PCA were developed to meet the same design considerations, such as displaying the scale of the data (fixed vs. adaptive) and selecting the plot type (scatterplot vs. heatmap). While linear regression is only able to display two variables in the statistical visualization, PCA is able to display as many variables as can be inputted given the computational constraints of the machine running the software, as well as practical visualization limitations due to space available for display, colors, and labels for variables within the display system.

Early development versions of PCA, created by a developer using Matplotlib in Python, show a point cloud visualization (Figure 15; Email Communication, 2/15/2018).
Later, after discussions with Environmental Team 1 and consultation of how PCA is typically visualized in statistical packages, line plots and variable labels were added (Figure 16). This led to a visualization of PCA where the first two components are displayed on the x- and y-axis, with a line describing the relationship between these two components. The PCA feature can generate as many components as variables included in the PCA, and also outputs a measure of explained variation in the data for each of the components. Additionally, numbers correspond to the eigenvector coefficients associated with the variables within these components, and are a measure of the loading, or the contribution that each of these variables has within this component, with higher absolute values indicating more contribution or lower absolute values indicating less contribution.

Figure 15: Early development prototype of PCA visual display
Summary of development and design

To develop features for linear regression, the team first used widely available statistical software to conduct linear regression analysis on data provided by Environmental Team 1. The results of this analysis were presented to the team lead, and
through multiple iterations and discussions, initial design specifications were agreed upon. Similarly, Principal Component Analysis was conducted on the same data, and a similar iterative approach was taken. After working features were developed, additional iterative contacts with Environmental Team 1 led to the further refinement of these features.

4.2.2.4 Implementation

At the end of the development period, the new VISTAS software features, described previously, were released through GitHub in the VISTAS repository as version 1.18.0. VISTAS is open source, so it is free to download and the source code is publicly available. At the time of our June 2018 group meeting that marked the endpoint of the development period, members from Environmental Team 1 had downloaded the newest version of the software with added statistical features but had not yet applied it within their workflows. However, during this development period in this final team meeting between developers and Environmental Team 1, the developers provided a short live demonstration of the new software statistical features and Environmental Team 1 commented about features they would like to see further developed, which were recorded. These recommendations included being able to manually set the value ranges for the x- and y-axis, choose color profiles in PCA scatter plots that mapped to the color profile on the landscape display, and fix bugs involving features unrelated to these the newly developed statistical features, such zonal statistics. The was also a discussion about integrating land use data types in the regression and PCA frameworks, but this was

---

3 The latest version of VISTAS available for download can be found here: https://github.com/VISTAS-IVES/pyvistas
understood as a topic that would need to be addressed in another iteration of development, as this feature would require changes to how data were imported into VISTAS, as well as pre-processing of data to delineate these categories before data was imported into VISTAS.

4.3 Post-period

4.3.1 Co-production

4.3.1.1 Context

Institutional factors and logistic factors

Collaborators for the VISTAS project were associated with three institutional types: academic, governmental agency, and research institute. The partnerships of these three institutional types, and their different funding sources and personnel within their teams, were typically directed by team leads who were principal investigators. Academic, governmental agency, and research institute affiliations of participants were separate in most circumstances, however, there were some participants in the VISTAS project who overlapped in their institutional affiliation. For example, a participant from Environmental Team 1 was associated with both governmental and academic institutes concurrently at the time of the project, and other participants employed by government or research institutes had affiliations and/or titles at academic institutions (e.g., courtesy appointments). Participants did not describe tensions between institutional types, although one of the developers from a research institute described how the process itself felt like it was influenced in part by institutional affiliation by being “more of an academic or research oriented project” and how they had “struggled” with understanding
this approach compared to their work on other projects (Development Team member D, 2018 interview).

Logistical factors are noted as important for facilitating successful co-production activities, as it is these interactions between participants, including the frequency of interactions, the quality of these interactions, and the context of these interactions, that set the stage for effective collaboration (Djenontin and Meadow 2018). A majority of collaborators on the VISTAS project were in close geographic proximity, with respective institutions located in the same city (Corvallis, Oregon), with some participants located further distances away (outside of the Corvallis area), but still within Oregon. One of the principal investigators on the project did not live in Corvallis, but would visit, in person, at critical junctures of the project. These critical junctures included such events as all-hands meetings that occurred intermittently (two to three times per year) when soliciting specific collaborator feedback about a feature or at a critical phase of development. More regularly structured interactions included weekly VISTAS Development Team meetings held via web video conferencing software, with interactions with non-developer collaborators occurring primarily by group emails, and, to a lesser extent, phone or in-person one-on-one meetings between developers and collaborators.

Participants did not identify logistical factors, at least regarding ease of collaborator interactions, as an impediment to the co-production effort. However, a member of the VISTAS Development Team discussed the role of collaborator proximity in a more nuanced way. While this developer was in close physical proximity, in terms of geographic distance, to the collaborators, this participant still felt comparatively far away, with closer proximity being co-located within the collaborator’s institution. From the
perspective of this participant, co-location in the same location could facilitate more “explicitly constructed and mediated interactions with [collaborator name],” feeling that the quality and the frequency of these interactions could be improved as a result (Developer Team member D, 2018 interview). In this sense, this developer would be located in the same facility as the collaborator for a certain number of days per week, and envisioned interactions where he/she could go down the hall and say:

“Hey, check out this thing, does that look like what you wanted?” For [collaborator name], when he found a bug, rather than having to send me an email, waiting for me to respond, or maybe not send an email but make a note to bring it up later, just walking down to the office and saying, “hey this thing doesn’t work” (Development Team member D, 2018 interview).

This developer also believed there could be a way to achieve this sort of interaction remotely, but it would require that these interactions were “supported at an institutional level. So if you are collaborating with other institutions then you don’t have a lot of control. But if that institution already supported people working remotely, you can get the same benefit without necessarily having a physical co-location” (Development Team member D, 2018 interview).

Summary of logistical and institutional factors

Participants in the VISTAS project were associated with at least one of three institutional types including government, educational, and research institute, with some participants having affiliations that spanned more than one institutional type. Logistics were not identified as a major impasse as many of the participants were in the same general geographic area, and were available to meet in person or through video conferencing software and communicate through email. One of the participants from the Development Team described proximity as institutional, rather than geographic, and
recommended that co-location of developers within an institute could be a model that would lower logistical barriers.

4.3.1.2 Inputs

Proficiency and expertise for knowledge co-production

Both the technical proficiency of the Development Team, and their collaborators, were frequently discussed by participants, especially with respect to the members of the Development Team engaged in the co-production effort. One of the developers directly discussed the expertise of the VISTAS Development Team, and how this expertise was leveraged when engaging collaborators. This included their own estimation of their personal expertise as one of project leads in co-producing software:

We have [Developer name] and [Developer name] who know a lot about visualization, how to make it work. I think what I know is I can listen to what [Environmental Team 1 lead] is saying, and I can kind of think about what is going to be useful for him. What is the vision? I have enough experience in developing software that I can say, well, I think we ought to start here (Development Team member E, 2018 interview).

Another project lead of the Development Team is not a computer scientist, but a social scientist, and brings a different sort of expertise to the co-production effort. This participant has a wide range of experiences working on co-production of knowledge projects that are not necessarily related to software development or visualization, including work tackling environmental- and climate-related problem contexts. One of the developers discussed how the contribution of this social scientist improved the co-production process, and proved an asset when discussions became more overly technical and risked losing sight of main project objectives:

Well there are a lot of times when [Developer name] will see through, we will be talking about technical stuff, and she’ll just see right through something and say, why don’t we do this? Or why are we talking about that, this is the issue…she just
has a broader view. And she’s done a lot of stuff in technology policy, and so she has a lot of experience with environmental science. And thinking about the way the public views things (Development Team member E, 2018 interview).

Another developer (Development Team member C, 2018 interview) discussed their experience working on interdisciplinary teams as “I’ve been interacting with non-computer science folks for a long time” and this experience helped them “recogniz[e] for a long time that they do not necessarily think the same way that computer scientists do.” While this was not necessarily experience in a more deliberately structured co-production activity like this study, this developer had exposure to collaborations with non-domain experts and had developed an intuitive understanding for how to navigate such interactions in the context of software and application development.

One member of Environmental Team 1 described their direct experience applying a formalized co-production approach that engaged multiple scientific disciplines and non-scientist stakeholders. This co-production effort was documented and written as an article that was recently submitted for review in an academic journal:

My background with co-production is a paper I wrote which is in review right now on co-production [of] decision-support tools between technical experts…computer scientists, ecologists, environmental scientists, and stakeholders. [The stakeholders] are community groups [and] watershed councils (Environmental Team member C, 2018 interview).

Other members of Environmental Team 1 also described their previous experience with collaborative research, with elements of a co-production approach. One junior member of Environmental Team 1 explained their work with a variety of different domain experts and stakeholders in the course of prior research projects, describing collaborative efforts that brought together “economists and other social scientists…health specialists, farmers” on large scale projects (Environmental Team member E, 2018
In these collaborative projects, Environmental Team 1 worked with interdisciplinary teams to “answer questions about ecosystem service and systems tradeoffs” with a focus on applying findings to “practical work” (Environmental Team member E, 2018 interview). This process, while not formalized in a co-productive sense, involved first identifying and defining a problem with stakeholders, gathering necessary expertise, conducting modeling and analysis with feedback from stakeholders, and then assisting stakeholders in applying this knowledge to a problem or addressing an area of concern.

**Summary of proficiency and expertise for knowledge co-production**

The Development Team had formalized expertise in co-production of knowledge, with members having experience in co-production applied to environmental management contexts and software development. The Development Team also had software engineers who had experience working collaboratively within interdisciplinary teams to create software and/or applications, although this was a less formalized co-production process. Environmental Team 1 echoed a similar sentiment, with some team members having direct experience with co-production activities with stakeholders, and other team members having less formalized experience.

**4.3.1.3 Legitimacy, trust, and inclusivity**

In post-period interviews, upon reflecting on the software development process more generally, some project participants discussed the conditions under which on-going collaboration effectively occurred, fostering a feeling of inclusivity in the co-production process. According to the lead of Environmental Team 1, part of the success of the project across a “ten year collaboration” was due to “the VISTAS team [being] very
inviting and welcoming feedback” (Environmental Team member A, 2018 interview). Such attitudes of positive interactions of collaborators with the Development Team were reflected by other participants in the VISTAS project.

The scientific lead of Environmental Team 1 discussed the level of accessibility to the lead developer on the VISTAS team, and the trust the participant had in this developer being willing to address the specific needs of his team. When asked about the direction of the overall project at the beginning of the collaboration, this team lead said “If there was anything that I felt should have changed I would have just mentioned it to [lead VISTAS developer] and we would have that discussion” (Environmental Team member A, 2018 interview). Another member of Environmental Team 1 mentioned that best practices in co-production were followed in the VISTAS project regarding establishing and building trust in the co-production process, noting there was a “focus on open communication, testing a lot of initial products then keeping communication lines open” (Environmental Team member C, 2018 interview).

A similar sentiment was echoed by the lead scientist in Environmental Team 1 with experience conducting collaborative and interdisciplinary research. This participant mentioned the critical role of a principal investigator from the VISTAS project in bringing together expertise, funding, and maintaining interactions and project activity over time. This collaborator stated:

I think [lead VISTAS developer] was good at bringing in a fairly steady stream of NSF [National Science Foundation] money which kept [VISTAS programmers names]. There were a number of people that kept the ball rolling and on our side we were able to keep a couple programmers going…it was sort of like a convergence that really enabled us to stay in touch and adapt VISTAS as we moved forward (Environmental Team 1 member A, 2018 interview)
When discussing potential explanations for this project’s longevity, both senior researchers indicated that it was due to key project participants with high levels of motivation and skill in engaging in collaborative research. In particular, the lead computer scientist of the Development Team and the lead scientist of Environmental Team 1 were identified as being powerful engines of continued progression of the VISTAS project across time. When commenting on the longevity of the project, a computer scientist on the Development Team remarked that “I think a lot of this is [lead VISTAS developer’s] influence, and I think its [lead Environmental Team 1’s] influence. That every answer brings about another question” (Development Team member B, 2018 interview).

When commenting on the experience of the VISTAS developer on other, non-VISTAS collaborative projects, and how this project may be unique compared to these other efforts, the participant reiterated this point:

I think it is largely the people. I think it is personality driven more than anything. Because the engineers [on other projects] could be in the same boat if they wanted to be. That every answer gives you another question. But the ones I was working with, they wanted something specific, and once they saw it in a visualization, they were good. That didn’t bring about any other question they wanted answered (Development Team member B, 2018 interview).

Some participants, while acknowledging the importance of project leads in sustaining and pushing the project forward, also acknowledged the importance of proficiency and expertise in generating successful outcomes. The level of technical expertise, according to one member of Environmental Team 1, varied across the life of the project. For example, this participant felt that the level of programming expertise fluctuated from development phase to development phase, stating:
Early in the project it seemed like there were a lot of PIs [principal investigators] and a couple of students who were not well trained were doing all the implementation. They never made any of their deadlines, and it was full of bugs, but it seems like this increment [current VISTAS development period] has been done much more professionally (Environmental Team 1 member D, 2018 interview).

This participant connected these deficiencies in implementation with the presence or absence of a software engineer on the VISTAS Development Team who, at times, had different levels of involvement throughout the project, stating “Honestly, things working well and not working well coincided with [developer name] being on the project or not being on the project (Environmental Team 1 member D, 2018 interview).” The technical input of this single member of the developer team, in the view of this participant, was critical to the successful development of VISTAS software.

**Summary of legitimacy, trust, and inclusivity**

Participants discussed the legitimacy that members of the Development Team brought to the development process, with a senior member being successful in creating trust in the collaborative process. This was contrasted against other participants’ experience with co-producing software, and without these key project members, these participants believed the process would not have been as successful in terms of outputs and longevity. One of the participants from Environmental Team 1 also identified the key role that one of the software developers played in influencing the quality of the development work.
4.3.1.4 Process

Setting-up and development

VISTAS software development was an iterative process, with initial goals and design considerations adapting to meet the changing needs for collaborators as well as the technical ability of the developers. The Development Team was unanimous in their support for such a model. One of the developers discussed how the co-production process impacted implementation of the VISTAS software, beginning at the start of the project “we did have an idea about what we wanted to do, maybe we didn’t all have the same idea” but then after multiple rounds of development “when that met reality and we started trying to solve problems for real people, it ended up being a different thing than what we had envisioned…we would have built very different stuff, obviously, if we had built software based on our original vision” (Development Team member B, 2018 interview).

Throughout this process, this developer, and other members of the developer team took the approach that they were “going to make sure we are building something that is useful for someone” rather than “have an idea of what we are going to build, and then we are going to build it, and then we are going to see if it is useful for people” (Development Team member D, 2018 interview)

A senior member of the Development Team (member B), when reflecting on the development process overall, compared it to other experiences of software development, and commented that they appreciated “how it has been so user driven.” This participant contrasted this approach with more top-down, developer-driven approaches, even if at times in the project there was not always a clear understanding about the feasibility of implementing some of the features desired by collaborators.
It is not a bunch people saying, well, I think we ought to do this now. It is a bunch of unreasonable users saying here is what we ought to do now. And so I like that kind of development. Especially I like it when users go, oh my gosh here is a pattern I didn’t see before (Development Team member B, 2018 interview).

However, developers thought this process of using co-production to create software could also be more challenging than other approaches to software development. Some of these challenges were associated with the various directions the co-production process pulled the software development process:

Maybe this is the nature of the project, and I’m thinking from a purely software engineering project point of view. It felt like there was a lot shifting focus with the project, and it always felt like we are looking at different things…I definitely had this problem where I just didn’t really have a clear kind of guiding start of this in the direction I’m going to head in (Development Team member D, 2018 interview).

This same participant reflected that “when it came to large goals, it just seemed difficult to really stay on task” and gave an example of VISTAS project work in machine learning (Development Team member D, 2018 interview). The VISTAS team pursued machine learning, specifically the application of convolutional neural networks to image analysis, with collaborators who had experience in the computer science machine learning domain. However, machine learning was never implemented in a release of VISTAS software, and, from the perspective of one of the developers:

I don’t think there was a clear understanding of how machine learning would work in VISTAS…I don’t feel that machine learning rose from so much of a need, as much as from a hypothesis standpoint. The idea that machine learning can help drive better science…rather than working on the project and one of our users comes to us and says, hey what I can really use is this, and we would say the best way to do this is machine learning. That is how I think of software being designed (Development Team member D, 2018 interview).

From this developer’s perspective, when the software development process strayed from the idea of “solve someone’s problem” to something that was “more a research question,
hypothesis driven” less was achieved “on the software side as would have if this was a pure software project” (Development Team member D, 2018 interview).

Another developer discussed how at times in the project the speed of the feedback cycle was slower than what was optimal from a programming efficiency standpoint, with gaps between programming sessions leading to additional time required to be reacquainted with the code when starting a new round of programming. This developer suggested that getting prototype interfaces to users with a faster turn-around would have been beneficial to the development process, from an efficiency standpoint, and for generating user feedback. In this developer’s estimation, feedback cycle speed, in particular feedback after the user had experienced a mock-up prototype, was critical, stating:

My feeling is that the first conversation is always, oh anything you can do for us would be useful. And they don’t really give you any useful information until you give them at least a mock-up of what you have in mind. And then maybe they will say, no that doesn’t help me at all. But more often they say, “you can do that?...Can you do this too?” (Development Team member C, 2018 interview)

When reflecting on the number of interaction cycles with developers that were critical for every stage of the development process, the lead from Environmental Team 1 stated that the overall effort was “a good model…I think there were frequent enough meetings to be able to adapt VISTAS and that flowed the other way too” (Environmental Team 1 member A, 2018 interview). When discussing the total number, or frequency of interactions, the lead Environmental Team 1 again reflected that communications were “the right level,” but also this participant felt “bad because I haven’t been able to keep up. My job has shifted a little bit. Someone told me once when I got into science that as you get older your administrative burden will go up. It’s about 60/40 administrative versus
actual work at this point, unfortunately” (Environmental Team 1 member A, 2018 interview). From this participant’s perspective, administrative burdens associated with his project duties were already at a high level, with a desire of being able to “hire more [hydrology modeling] team members” and acquire more resources to leverage toward his modeling efforts and the VISTAS collaboration (Environmental Team 1 member A, 2018 interview).

Another researcher from Environmental Team 1 mentioned that one benefit to the way VISTAS software development had progressed was that VISTAS provided “beta projects that then the ecologist can iterate on.” During the development process, this researcher also noted that “communication channels were open, there was constant feedback” and that this process was a very positive experience (Environmental Team 1 member C, 2018 interview).

As part of the co-production effort between Environmental Team 1 and VISTAS, one of the Environmental Team 1 members thought that the co-production effort did not fully utilize the skill of Environmental Team 1’s expertise, especially when it came to the implementation of the software and promoting good practices. This participant recommended that the collaboration could have been a “little bit tighter of a partnership, there was a long period before we ever saw any of the code. And I think that a lot of bad practices happened that maybe could have been prevented,” suggesting that the Environmental Team 1 had “skills to offer that were never used, that seems like a bit of waste” (Environmental Team 1 member D, 2018 interview). This same team member, when reflecting on the development process overall, suggested that future efforts should always strive to “be realistic,” stating “There are points in the project where I think they
were trying to do too much. Or shooting too high. They ended up compromising the work that they were actually able to get done” (Environmental Team 1 member D, 2018 interview)

Another researcher on Environmental Team 1 discussed some of the inherent challenges of working on a co-production project, or collaborative projects more generally. From this participant’s perspective, one of the negative aspects of working in a group is that the “squeaky wheel gets the grease. So whoever is able to put forth the most salient argument they win. And then software development runs in that direction” (Environmental Team 1 member C, 2018 interview). An example of a software development direction that was less relevant to the participant was the implementation of Principal Component Analysis:

Someone would come up with the idea, PCA is useful for me, right? And for whatever reason, that won out in the discussion. So then the software development went towards PCA. And then meantime, I’m a fly on the wall and I think PCA is not useful for me at all. So why are you guys spending all this time doing this? (Environmental Team 1 member C, 2018 interview)

**Summary of setting-up and development**

Participants in the VISTAS project generally agreed that the co-production process was an appropriate approach for setting-up and developing software, and that it fit the needs of participants. Participants also acknowledged that this approach was likely more time and resource intensive compared to other development methods. Some criticisms of this approach were associated with the management and application of expertise within the team and the frequency of group interactions. One member of Environmental Team 1 thought that the expertise from the scientific team could have been applied more effectively throughout the development period, while members of the
Development Team described how software iterations involving new software features could have benefited from a faster feedback cycle.

4.3.1.5 Implementation and Outputs

When referencing the co-production effort across time, those participants who had been involved in the project for an extended period of time, and who had experience with other collaborative projects, noted that the VISTAS project had a longevity and level of commitment that surpassed other projects they had been a part of. For example, a senior computer scientist on the Development Team noted:

I’ve been involved in smaller things [co-production projects], like where some engineers have some simulation data that they want to verify as doing what they expect it to do. And in that case it was we will just whip together a program that shows them some things and call it a day. But VISTAS is a project that has been going on for what, seven years? So that right there sets it apart from anything I know of (Development Team member B, 2018 interview).

Another developer discussed how the implementation of the new software features met the expectations of the project collaborators, with the opportunities to extend the features with more time and budget:

It seems like we met the basic components of [new VISTAS features]. If we had another year and budget to work on it with, there is a lot more for us to do. I could see doing more statistics and shaping the interface that we have for the linear regression and multiple regression and PCA. I’m not sure that there is more that we would add to those components themselves…I think that we have a good initial implementation that is useful to people (Development Team member D, 2018 interview).

This same developer (member D, 2018 interview) did not have much confidence in the continued development of the software, mostly because this developer had experience with maintenance efforts of software projects with similar user-bases being unsuccessful, stating “at [research institute] we are always having people trying to hand [software] off to
us saying ‘Hey will you guys maintain this thing with no budget?’…no.” This developer did believe that if there was “a community around the software…with hundreds of scientists using it” this would be much more likely to be maintained by academic institutes or people with programming knowledge in the user community (Development Team member D, 2018 interview).

When discussing the implementation of software, one of the lead software developers, reflecting on what they thought the output of the software implementation was, described it in three parts (Development Team member E, 2018 interview). The first outcome “solving [Environmental Team 1’s problem] of visualizing their data. They now have a tool they can use, and if the tool falls by the wayside they know what they want.” The second outcome is that “[developers] as researchers into visualization usability and scientific visualization have a better idea about what is a useful system.” And third, “that visualization without analytics isn’t enough…you need to combine those things and link them together so that you can do them seamlessly…we are getting closer [to achieving] this.” In the estimation of this developer, “if there was going to be a next proposal” it would focus on extending the linkages of visualization and analytics into other scientific domains and finding a way to lower barriers to such development (Development Team member E, 2018 interview).

All of the participants in Environmental Team 1 viewed the co-production effort between the VISTAS Development Team and their team as a success, with differing explanations for why it was successful. For example, a member of Environmental Team 1 viewed this development as something their team would not have had the resources to implement themselves:
I think [VISTAS] is a success. [Environmental Team 1 lead] is very pleased with the kind of images that [they get]. I think that it furthers [their] science and we couldn’t have been funded to create a 3D visualization program so it has been very synergistic (Environmental Team 1 member D, 2018 interview).

Reinforcing this sentiment, another researcher on this team described how the resulting product was a success, but part of the usefulness and longevity of the tool is dependent on its long-term maintenance, which, at the time of the interview, was uncertain:

It’s a useful tool, [Environmental Team 1 lead] loves to make images and look at them, he uses them to make slides, he uses it to put in papers, so it is a tool that he is going to continue using. The choke points are always if something changes out from underneath it (Environmental Team 1 member B, 2018 interview).

They further discussed how the decision to rewrite VISTAS software from C++ to Python was one way to increase its longevity, even though this rewrite did not outwardly result in many feature improvements at the time it was implemented. Even with the Python rewrite, however, this researcher discussed how this may not be a guarantee of longevity:

It’s all written in Python, but if Python changed incompatibility then it wouldn’t run anymore, then that would be a problem. I think they have future proofed it a little bit by making it in Python, but Windows could change, something else could change, in which case we’d have to then weigh how much it would cost us to fix it vs. the utility (Environmental Team 1 member D, 2018 interview).

However, when describing the recent feature development of VISTAS, one of the researchers from Environmental Team 1 discussed how they thought the core purpose of VISTAS was visualization, and the addition of statistical and other analytical features were unlikely to be beneficial to his work, while acknowledging that these features could likely be useful for other members of his team:

So VISTAS is valuable because it is able to visualize in movie form, ASCIIIs. That’s it. Statistical stuff, PCA (Principal Component Analysis), does not matter to me. It is just can you more simply, more quickly, develop movies for me and
my ASCII data. That’s enough for me (Environmental Team 1 member C, 2018 interview).

When discussing whether the latest development period was effective in meeting the needs communicated by the group, with specific reference to visual analytics, one of the participant’s from Environmental Team 1 saw these results as mixed, stating, “Yes and no” to whether the VISTAS project had achieved the goals established at the beginning of the development period:

Yes and no. At the end of the last period, there was this meeting where we discussed where to go from here. And we had a whole bunch of suggestions, mostly they were [eco hydraulic model]-centric, and it seemed like what [VISTAS Development Team lead] wanted to do is to go big and write a big NSF [National Science Foundation] grant about machine learning. And I guess [they] didn’t get the grant...and it sort of morphed into a much more modest set of goals. Given that you were going to do statistics, and given that [Environmental Team 1 lead] liked the set of statistics you did, I think it suited our needs. But it seemed like the original directions that we went for, which I don’t remember what they were anymore, were lost (Environmental Team 1 member D, 2018 interview).

**Summary of implementation and outputs**

Developers and members of Environmental Team 1 described the results of co-production as successful, citing the use of the software by their team for both understanding and communicating their data, and the software’s role in facilitating and advancing their research. The new VISTAS features, the result of this development, had not been applied in current workflows, but members of Environmental Team 1 described how they intended on employing these features in future work. Some of the environmental scientists and developers expressed concern about the longevity of the software, even though actions had been taken to make it easier to maintain.
4.3.1.6 Outcomes and Impacts

Changes in practice

A researcher on Environmental Team 1 discussed how working within a collaborative effort such as the VISTAS project changed the way this researcher viewed the process of conducting interdisciplinary research, and also commented on other—perhaps less successful—efforts in collaborative work, at least with respect to other projects in their field:

That’s been one thing about VISTAS, the team specifically has been great…I think I’m kind of amazed. I’m hearing this from outside realms that other disciplines are trying to get to the point where you are intermixing expertise. But then I don’t know why other disciplines are having so much trouble with it. But working on the [Environmental Team] and the VISTAS team, I just think we get a hell of a lot more done when you have a bunch of experts who have completely different backgrounds, you just have lots of different perspectives (Environmental Team 1 member B, 2018 interview).

This same collaborator from Environmental Team 1 discussed the value of interacting with a multidisciplinary team, and how this process involved communicating to other team members. Through this communication process they felt that they had improved their understanding of their own work through a process of reflection, but also better understood other’s work. For this participant, it was the act of explaining their research that was most impactful:

I think explaining your work helps solidify it. And, for me, helps me understand it more…that is what the collaborations do is they make us explain. As we are pushing research forward and doing new things that no one else has done…we go pretty far ahead and then not really reflect. But working in these groups with other people who are pushing the envelope forward in their own right, but then you have to explain what you are doing, then that helps reflect and summarize your thoughts (Environmental Team 1 member B, 2018 interview).

The impactful nature of communicating science to other non-disciplinary experts also contributed to the participant’s understanding of the limitations of their work, a product
of this reflection process. Speaking about the role of communicating science within this co-production effort, another participant noted that:

It also helps people understand the limitations. We get to this point where we think everything that we are doing is great, and solving every problem, but then you go back and summarize and try to explain to someone else and then they start asking you questions about it, then limitations start to get revealed, which is also quite important (Environmental Team 1 member C, 2018 interview).

For a member of the Development Team who already had substantial experience working with other non-developers, such as scientist collaborators, this co-production effort did not represent a new experience, but reinforced the perspective that such approaches to software development were beneficial, especially when compared to other software development efforts that this participant was involved in:

In my current position, I’m in a group where we tend to have this very narrow pipeline of information between users and developers. Which I find frustrating. I don’t think this helps with making good, usable software. I think we need more communication between developers and users, and the sort of meeting with [Team Lead of Environmental Team 1] and his group have sort of brought that home…Where you have frequent face-to-face meetings with users where they are telling you, “No that isn’t what I want. Here is what I need to be able to do” (Development Team member C, 2018 interview).

When asked how their involvement with the co-production process may have changed the way that they view the software development process, this same member of the developer team, who was only recently involved with the project, said the process overall “fit within the framework of how these things should work” with respect to “interacting with users, philosophy about user interface design” (Development Team member C, 2018 interview).

A lead developer reported that their approach to software development in the environmental science domain had changed though their involvement with the project. Rather than rely on the expertise of computer scientists to be involved with the
development of new software for domain scientists, how can the lessons from such co-production activities “make it so scientists can do more of their own development…without having to go through a programmer” (Development Team member E, 2018 interview). Other changes in practice were associated with how participants approached work with others, regardless of whether this work was related to co-production, but collaboration more generally. For example, a member of Environmental Team 1 described how in research settings domain scientists working in collaborations with non-domain scientists “forces you to explain what you are doing…and explain to someone else and then they start asking you questions about it then limitations start to get revealed” (Environmental Team 1 member B, 2018 interview).

This developer came to the realization that computer scientists and environmental scientists do not frequently collaborate on projects, such as VISTAS, because what “environmental scientists need help with are of a research nature, so it is not obvious what a computer scientist would get out of this” (Development Team member E, 2018 interview). Additionally, this developer acknowledged that “any collaboration is high risk. Sometimes it takes a year of talking to somebody before you find you’ve got something.” From this developer’s perspective, successful software co-production between computer scientists and environmental scientists requires conditions such as:

- Time on the part of the computer scientists to involve themselves in the project and the environmental scientists would have to be willing to spend time with the computer scientist and will have to trust that they will get something out of it like a usable system in the end (Developer Team member E, 2018 interview).
Summary of changes in practice

Both developers and members of Environmental Team 1 discussed how the co-production process in the VISTAS project either changed their perspective about collaborative work or reinforced its importance. Of particular note is that such a process encourages interactions that result in discussions about adjustments of development goals and candid discussion of limitations. Participants also described that if given the opportunity they would seek out projects that involved cross-disciplinary and/or stakeholder collaborations, while a member of the Development Team noted that co-production efforts are higher risk, especially for computer scientists.

4.3.1.7 Salience of knowledge co-production

Many of the participants reflected on the relevance of the knowledge co-production effort for software development projects, including, but not limited to, their experience with the VISTAS project. As discussed in some of the previous comments, developers described how the process of co-production might have been slower than other approaches, but this approach resulted in a product that ultimately was useable and useful, and how the co-production process facilitated this:

To some extent, [co-production] was perhaps a slower process [compared to other software development]. I think what we created, via co-production, was a much more tangible, understandable, and meaningful product that was able to actually go off and solve problems, instead of, “I need a thing that does three apps. Can you do that?” (Developer A, 2018 interview).

Some of the researchers reflected that the co-production effort did not achieve all of the goals proposed at the start of the project and in funding applications. However, many of these goals had shifted based on available resources, the expertise of the Development
Team, and the changing needs of the participants. Project outputs not related to the software product were also recognized as important and noteworthy. One former member of the Development Team in a previous iteration of the project summarized this sentiment as follows:

I think the hope was that it was going to be this crazy robust tool that is going to be open source, and magic was going to happen. And I think the magic happened when all these people came together. And maybe it was good that we had a huge turnover of tech people because they got exposed to the scientists. And there is a chance to kind of integrate all of the people who normally may not have had a chance to have a conversation (Domain Expert F, 2018 interview).

This same participant reflected on how the actual resulting knowledge production was more than the creation of an output, in this case a software tool, and instead was about the building of relationships and acquisition of behaviors to generate this knowledge. When discussing how the co-production process influenced research activities, Domain Expert F stated:

I honestly feel like the actual process of them doing it was more important than the actual output of the tool. I mean, talking through how to build something together is more important than what is actually built, in my opinion, right now, in this place and time (Domain Expert F, 2018 interview).

A senior member of the Development Team discussed how the outcomes for stakeholders in co-production efforts are different, because the participants themselves are different. In this respect, whether project outcomes were met, and such a process was effective in achieving these outcomes, could viewed very differently given the perspective and domain of the stakeholder:

Co-production depends on what are the desired outcomes for each of the stakeholders, right? So, it is not less of a co-production to have a research project where the outcomes are computer science research as well as environmental research, it just means that the outcomes are different than if they are the only desired outcome in the co-production (Development Team member E, 2018 interview).
When discussing pursuing co-production as a development approach, this same developer described how such activities may not always have traction given the challenges they present, especially considering how much time they require of participants. Such interactions may not necessarily result in outcomes that further their science, or produce outputs that fit within traditional academic outcomes.

Any collaboration is high risk. Sometimes it takes a year of talking to somebody before you find that you’ve got something…for us, we developed a relationship with [Environmental Team 1 lead] over a very long period of time…. [Development Team lead] hasn’t been as involved as [they haven’t] seen something very specific [that they] and [their] student can get out of this. [A different Development Team lead] saw something that was relevant to [them]…I’m more interested in the software engineer aspects of the project, and the user experience, and for me, I get that out of it…also, I’m toward the end of my career and if I don’t get something publishable that is sort of OK…[Environmental Team 1] gets a system that [they] need (Development Team member E, 2018 interview).

A member of Environmental team 1 also acknowledged this challenge of engaging in co-production work, saying that before they had begun the process they were unsure of such a collaboration, but after their involvement with the project they would be willing to participate in projects under similar conditions. However, they also acknowledged that participation in co-production could pose risks:

I didn’t have a negative view of it, just had a sort of unknown view…anything I worked on in the past was with a team of people [where] their educational background was more similar. I’d be very open to [co-production] after working with the VISTAS team. I would hope that this new team would have as good as people as the VISTAS team. That’s a risk no matter what (Environmental Team 1 member B, 2018 interview).

None of the participants reported that the co-production process was not relevant to their work, or that based on their experiences with this co-production effort, they would not pursue similar collaborative work in the future. Among the participants interviewed, they were unanimous in their agreement that the co-production approach
was appropriate for the problem domain, and that similar co-production of knowledge research projects should be supported in the future.

**Summary of salience of knowledge co-production**

Participants acknowledged that while the co-production process was likely more time consuming and higher risk than other approaches to software development, the resultant product and experience was useable and meaningful. Some participants commented that the process of creating the product was more important than the product itself, with other benefits including exposure and collaboration with other disciplines. While outputs from the project may not have always met traditional academic outcomes, the desired outcomes of stakeholders were met, and this positive stakeholder outcome provided support for the co-production effort overall.
5 Discussion

5.1 Visualization needs and software design

5.1.1 Challenges with data

The findings presented in the previous chapter are consistent with research conducted by scholars exploring the role of visualization in environmental sciences and ecology, suggesting that scientists in these domains often face substantial challenges associated with data and/or data visualization (Cushing et al. 2012; Winters 2015; Winters et al. 2016). For participants in the VISTAS project, challenges identified with data were multifaceted. Many of the challenges described by participants were associated with the type of the data that were being visualized and were comparable to attributes of big data more generally, sometimes characterized as the Vs of big data, including veracity, variety, volume, and velocity (Chen and Zhang 2014, p. 314; Constantiou and Kallinkikos 2015).

Of these three Vs, data descriptions aligning with veracity and volume tended to be most prominent among our participants, however, some members of Environmental Team 1 and 2 explicitly mentioned the speed of incoming data from modeling being problematic, with participants lacking the capacity to easily manage new data outputs and understand or interpret these outputs with the expediency necessary for their workflows. Challenges related to the volume of the data were described in terms of the size of outputs generated by models, which in some cases were terabytes, a volume too large to be easily applied in conventional data workflows, often requiring scientists to use sub-setting or filtering techniques that reduced the size of data to be more manageable. This
sub-setting of data left some participants concerned that patterns and relationships in their data were being left undiscovered, particularly so when their data had both high temporal and spatial complexity—a common theme articulated by our participants who did modeling at landscape scales using alternative climate and policy scenarios.

The veracity (i.e. trustworthiness) of the data also presented challenges, which was discussed primarily with respect to data uncertainty, and, in particular, how this uncertainty was measured and then applied in analysis. Since both Environmental Team 1 and 2 used real-world data to calibrate their models, which introduces uncertainty, and then develop models based on this data, which also adds to this uncertainty, parsing and representing such uncertainty can be challenging. To address uncertainty in their modeling exercises, Environmental Team 2 described using simulation approaches that generated confidence intervals or measures of output variability.

These challenges, discussed by respondents in the context of their domain, are similar to issues faced by other domain scientists who generate modeled data using complex simulations (e.g., Chen, Chiang, and Storey 2012; Pijanowski et al. 2014; Viceconti, Hunter, and Hose 2015). While participants described these data challenges associated with the modeling process and acknowledged that current approaches were often inadequate, some described data visualization as one potential technique for helping address such big data issues. Other scientists, however, from both Environmental Teams and Domain Experts, did not find visualization to be particularly helpful for addressing issues related to big data in their workflows. It is difficult to speculate as to why this is the case for these participants. It could be that visualizations were not seen as helpful because of the substantial challenges associated with creating useful visualizations, or
that data visualizations were not helpful for conducting their science. However, it is also possible that there are visualizations that could help them better understand their science, but these scientists had not yet been exposed to visualizations that facilitated this understanding. If this is the case, suitable visualizations may be unknown to some scientists, and only through a process of testing and exploring different visualizations can the appropriate visualizations that can help make sense of these scientists’ data be discovered. This suggests a need for approaches that allow for flexibility and iterability in the visualization generation process, which presents its own set of challenges.

5.1.2 Challenges with visualization

Similar to the data challenges associated with big data-like characteristics, there were also considerable challenges associated with data visualization, with these two themes frequently mentioned together and sometimes interchangeably. Indeed, there is an expectation that challenges with data and challenges with data visualization are connected since it usually requires some understanding of data structure in order to generate a visualization of the data. While the big data issues described above are not necessarily addressed through visualization, participants identified visualization as a way to generate new insights about data, as well as communicate insights about data to others.

For environmental scientists and ecologists in the VISTAS project, visualization challenges were related to the production of visualizations using software. Visualization software that participants used were often widely available commercial software, either proprietary or free and open source, and commonly applied by scientists within their discipline. Some of the identified challenges were associated with software ease-of-use, with generalized software platforms having so many available features that it was
difficult for participants to identify which features needed to be applied to generate a desired outcome. Other challenges were associated with a software platform’s functionality. Either participants were unable to create the visualization appropriate for their data, or, if they could generate such a visualization, this visualization was too slow to be created in real-time, making it difficult to generate new visualizations of outputs in workflows that required multiple iterations of “tweeks” to the model, or feedback from other scientists or stakeholders. Still other challenges were associated with literacy and knowledge related to appropriate visualizations for participants’ data, with a need for domain software that helped scientists better identify and isolate the phenomena of interest. These challenges established a need for visualization software that could address such issues while maintaining usability and being useful for completing tasks such as exploring data and communicating to scientist and non-scientist audiences, key issues that participants identified as critical components of their science.

One of the central visualization tasks that emerged for understanding data outputs for Environmental Team 1, and to a lesser degree Environmental Team 2, was the display of complex topographical information. For these scientific teams, the research phenomena they were examining were both spatial- and elevation-dependent, making it particularly challenging to interpret data outputs using two dimensional representations (i.e. bird’s eye view). For display of spatial data, software typically applied was Esri’s ArcMap, which, in the way our participants described their use of it, did not fit their needs for producing visualization of complex topographic data at landscape scales.

As a way to effectively display topographic information, a 3D representation was identified as useful for these domain applications because topography was often critical
for characterizing the phenomenon that the scientists were studying. For our scientist collaborators, this phenomenon included the dynamics of water on the landscape, such the movement of water within a watershed or a nearshore coastal area. While these findings were not surprising given that one of the modes of inquiry in the original VISTAS project proposal resulted in the development of software capable of presenting topographic and temporal data concurrently, it supports the software development approach that was undertaken in the VISTAS project that made 3D topographic display a focal point of the project.

Another key visualization task identified as important for understanding modeled outputs from Environmental Team 1 and 2 was the representation of temporal or time-based data on visual displays. Adding a time dimension to any visualization presents unique challenges, and for visualization of topographical data, these challenges are intensified. Time was an organizing factor for much of the research activities described by the participants because modeling activities were simulation-based and often considered the change of spatial variables across the timescale being modeled. Displaying spatiotemporal data in a way that can be easily interpreted is an ongoing research effort, with a proposed solution being the creation of temporal volumes (e.g., time extrusion). Members of the Development Team working with such temporal volume visualizations thought this level of abstraction may be too high, particularly so for the communication of model results to non-scientist audiences.

However, there was still a desire expressed by participants to glean more from their data than just spatial and temporal display, with scientists seeing a potentially powerful role for data analytics being incorporated in visualization tools. While
participants were generally unsure which analytics tools should be included, they
described analysis that varied by application, ranging from statistical methodologies to
machine learning approaches. Overall, there was a desire to be able to use such tools to
find interesting features in the data, which some participants described as hotspots. Given
the complexity of the modeled data that these scientists were working with, alongside the
domain-specific needs of scientists, data analytics features would likely need to be
specific to the tasks of the scientists, especially because participants described how such
data analytics features were not usually included in visualization software that was more
generalizable and in wide use.

Another dimension of the visualization process was the communication of results,
both to the public and to other scientists. Some participants saw visualization design for
different audiences as being important for helping these audiences understand their
science. However, others worried that visualizations, especially visualizations with more
sophisticated design elements, such as 3D fly-through videos, as being too convincing to
non-scientist audiences unfamiliar with the domain, and that these audiences were less
likely to question the underlying data. For presenting to scientists an opposite effect was
observed, with domain scientists having an expectation about the types of visualization
they would see, and deviations from these expectations sometimes made it challenging to
communicate visualized results. Still other participants did not make strong distinctions
between scientist and non-scientist audiences, and instead emphasized creating
visualizations that could be understood by anyone viewing them, with or without training
or expertise.
While the participants who were interviewed in the pre-period all described a need for software capable of displaying data in 3D, their ultimate intended use of this software varied, with some discussing how 3D temporal visualizations either were currently being applied, or would be applied, to both understand and communicate model results. Somewhat surprisingly, we discovered that such visualizations tools did not only have applications for understanding data, but also for validating models. While the distinction between understanding and validation may at face appear inconsequential, participants made this distinction with respect to the process of testing whether model integration efforts resulted in outputs that were consistent with features on the landscape. In this respect, the validation process was more iterative than simply understanding data outputs, as the process of integration led to changes in modeling that resulted in different data outputs, that could then undergo another iteration of validation.

5.2 Development of visualization software

Considering these themes described in pre-period interviews, there was a demand among our collaborators for visualization tools that are easier to use, more specific to their scientific domain, have the capacity to display both temporal and spatial data, and can perform data analytics. Given that the current software market has either failed to provide a product that meets this need or is not capable of producing such a software product that meets these needs due to the domain specificity of desired features, a different approach is required to meet the needs of the collaborators in our study. Of particular note is that while there may be software platforms capable of performing the visualization tasks described above, this software is either unknown to VISTAS participants or is not considered sufficiently useable from a user experience perspective.
for our collaborators to apply in their workflows. Since these challenges surrounding the production of visualization software are strongly linked to a scientific domain that addresses complex environmental problems using models that produce big data, traditional approaches associated with software development are likely insufficient, or are a poor fit for addressing such needs in plan-driven development approaches. Even software development approaches with more flexibility, such as Agile Software Design and User Center Design (UCASD) (e.g., Silva et al 2011; Brhel et al. 2015), while proposed as a way to quickly respond to changing user needs and preferences, does not necessarily fit this need when participants are not sure what their initial needs or preferences are. However, addressing such needs are within the purview of a different approach, co-production of knowledge, which we applied to the VISTAS software development process.

While the pre-period inquiry reinforced a core focus of this project, the co-production of visualization software for displaying complex environmental and ecological data, it is important to note that the future direction of the development period was largely unknown at the time of the pre-period interviews, making even less traditional and more flexible development approaches such as aforementioned agile methods challenging to implement. Although the developers and the scientific collaborators agreed there was a need for development of visualization software, the scientific collaborators either could not always easily describe what this need was, or found that their needs changed over time. An example of this, presented in the development period findings, was the development of data analytics, which began as way to gain additional insights from visualized data, and was an agreed upon development
goal after multiple iterations between scientist collaborators and developers. However, when the Development Team asked what sort of data analysis tools should be developed, it was only after another set of multiple iterations with the developers before the scientist team and the Development Team converged on the development of two specific data analytics features: linear regression and principal component analysis.

Since the scientific team had not conducted principal component analysis or linear regression on their modeled data before, they did not know whether these techniques would be useful or useable (i.e. applicable to their data and their domain). The lead scientist had not seen such analysis run on modeled data until the members of the Development Team presented results from these analyses using statistical packages. Even before such analytical features were added to the software through a development iteration, scientists who saw the outputs from these analyses began asking new questions about their data, their modeling, and the outputs from their models.

After developers and scientist-participants agreed that linear regression and principal component analysis should be included in the software, and that these analytics should be linked with the visual display of data, it was again not until they saw examples of proposed visualizations associated with the statistical features could they provide feedback. Interestingly, much of this feedback in the initial interactions between developers and scientists was not associated with major changes to visual displays, but how the existing display was formatted, with more minor suggestions associated with configuration of x- and y-axis, for example. After these display configuration issues were identified and resolved, more complicated features, such as inclusion of nominal categories for these visualizations, were broached with the Development Team. Because
the more complex additions were mentioned near the end of the development cycle, when much of the software developers time on the project was already expended, these features were not addressed in this most recent round of development that was the focus of this research. As noted by the team, if there was a second round of development it would likely include building out this feature set.

This process of software development had characteristics of agile and user-centered approaches, such as UCASD (Silva et al. 2011; Brhel et al. 2015), with a few key differences. In UCASD, usually the collaborating user group has a clear idea about the general specification that they want, and the feedback cycles are structured around this specification/goal. However, in the software co-production approach that we followed, it was immediately evident that while the participants had some idea about the overall direction they would like to see the software take (i.e. data analytics), they did not know what set of features they wanted. Only when presented with outputs from analytical techniques applied to their own data, and given time to formulate scientific questions that arose from this analysis, did our participants begin to clearly articulate desired features.

First, feedback on requested features were focused on the details associated with its existing functionality and display, and after these were met, feature requests became sophisticated and aspirational but still within the context of data analytics. It is the lack of an initial specification and willingness to engage in the production of science that represents the greatest contrast between the approach we take in the VISTAS project and a design approach that follows a user- and/or agile-oriented software development framework. However, that is not to say that this co-production process could not be improved through incorporation of agile principals. Indeed, some of our participants
indicated that a faster feedback cycle was desirable, and while it is a recommendation of agile design to have short design cycles with incremental improvements (Brhel et al. 2015), this puts more boundaries on participant interactions than what is typically articulated in co-production literature (Lemos and Morehouse 2005; Visbeck 2008; Ziervogel et al. 2016; Wamsler 2017).

Some members of Environmental Team 1 thought these data analytics features were a useful addition to the software, while others were less certain about the usefulness of these extended features and thought that new development should have instead focused on improving existing visualization features. However, all participants were unanimous in their support for development being conducted using a co-production approach.

5.3 Co-production

5.3.1 Inputs

Members of both the Development Team and Environmental Team 1 had experience working in interdisciplinary teams that collaboratively produced knowledge, but the context and formalization of this process varied. Two members of the Development Team had direct experience with co-production, one with applications to computer science and software development, and the other to environmental management problem domains. Other members of the Development Team had less formalized experience with collaborative development, which included iterative feedback cycles in client-developer relationships. For Environmental Team 1, all members had experience working on teams that included other scientific disciplines as well as diverse stakeholders such as land managers and farmers. One of the members of Environmental Team 1 had direct experience with co-producing decision support tools. Such experience with
collaborative research, and co-production more specifically, may be one reason why some members of both the Development Team and Environmental Team 1 were initially supportive of the software co-production process, as they already had experience working in this framework and had observed its outcomes.

Another reason for the willingness of Environmental Team 1 to engage in the co-production process is that they had trust in the process and viewed it as legitimate. Participants described how previous interactions with the Development Team were always inclusive and fostered an atmosphere that was open to feedback. More specifically, trust in the process was associated with the involvement of a few key members of the Development Team who generated trust in the process, in particular a lead researcher’s ability to bring in a steady flow of funding and the participation of an experienced software engineer. Meadow et al. (2015) discusses the importance of key stakeholders who are critical for establishing channels of communication and the deliberate relationship-building between participants. These crucial stakeholders act as “an intermediary between users and scientists” facilitating two-way communication and the production of usable science and the development of long-term relationships (Dilling and Lemos 2011, p. 685). In addition to these key participants, there was also a software engineer on the Development Team who had been involved for the duration of the entire project, spanning seven years, and was recognized as an expert in the domain of visualization software, and whose involvement in the project was seen as essential and connected with successful feature implementation at various stages of the project.
5.3.2 Co-production development and process

When reflecting on the setting-up and development of the VISTAS software, participants noted that the co-production approach that was taken appeared to fit the needs of the participants. However, nearly all participants noted that this co-production process was more time consuming than other software development approaches. Specifically, one of the developers noted how the VISTAS project was an unusually long collaboration. This developer was more accustomed to software development efforts that were much shorter and had concrete goals and endpoints. It is challenging to compare the development approach of VISTAS software with more traditional, plan-driven software development approaches because the VISTAS project went through multiple development iterations where substantial feature development was undertaken. The intensity and involvement of the team and participants across the life of the seven-year project also varied. A more appropriate comparison between the VISTAS co-production project, and other software development project, may be to examine a single development period. For example, the most recent development period for VISTAS lasted approximately seven months and is commensurate with timescales of software development projects using user-centered and/or agile design.

Given that the approach of the VISTAS project toward software co-production was considered to be time and resource intensive by participants, what drove their continued participation? While such a question is challenging to answer given the lack of a counterfactual, such as the same group of people engaging in a different development process, many of the participants reflected that they appreciated that the setting-up and development process lead to interactions where “every answer brings about another
question” and there was no push to establish a project end-point that limited the scope of exploration (Development Team member B, 2018 interview). However, such an approach to development was also identified as generating problems, with criticisms about personnel expertise and management in early iterations of the project, and in the current development period the need for more frequent and explicitly constructed feedback cycles. Such feedback cycles are more associated with formalized approaches to software development such as agile methods or user-centered design. Other criticisms included not effectively leveraging the computer science expertise of members of Environmental Team 1 in the development of VISTAS software process, which also indicates there was domain-specific knowledge of collaborators that could have been incorporated in the co-production process.

Even with such criticisms, participants reflected positively on the co-development process overall, and, in particular, its flexibility and fluid nature, which was seen as both a positive and negative aspect of the development phase. This seems to suggest that if it were possible to maintain fluidity in the development process (e.g., “every answer bring about another question”) while imposing a more formalized schedule for feedback cycles, such as advocated in agile methods, a hybrid approach incorporating more regimented interactions could be an effective way to conduct co-development under this context. However, it is also possible that imposing a more strict feedback cycle may, as a consequence, result in developers and collaborating scientists setting explicitly defined goals and being less likely to deviate from such paths in the development period, which would detract from a key feature of this development process and the process of co-production more generally.
5.3.3 Outputs

Similar to how participant feedback about the VISTAS software development process was, on the whole, supportive of this process, participants also had positive assessments of the resulting product from this co-production exercise. This positive assessment of VISTAS software is likely associated with the newly developed features being a product of an inclusive process that sought to meet participant needs and new software features either meeting, or partially meeting, such needs. Such reported outcomes related to the usability and usefulness of co-produced knowledge is a noted output of successful co-production activities (Lemos and Morehouse 2015; Djenontin and Meadow 2018). It is unknown whether the participants view VISTAS software as useful and useable because they guided the development toward new features that met their needs or because participants were involved with development they are now biased towards perceiving features as useful and useable even though their needs are not being met. Indeed, both could be true, with the linkages between usability and usefulness requiring some combination of stakeholder involvement and knowledge production that fits the needs of these stakeholders while generating perceptions of proprietorship. In this respect, this research reinforces the value of participants having tangible stakes in the development of the product. Creating buy-in and including stakeholders in the knowledge production process is seen as a critical step for addressing complex problems and producing scientific knowledge (Funtowicz and Ravetz, 1993; Lemos and Morehouse, 2005; Dilling and Lemos, 2011; Meadow et al., 2015).

Since the output of this co-production activity involves a collaboration of many participants with diverse backgrounds, needs, and perspectives, the co-production process
does not necessarily meet all the needs of all of the participants. One of the participants in Environmental Team 1 noted that they saw little value in developing data analytics features, and instead would have liked to see more development of the topographical visualization display. For this participant, the core functionality of VISTAS software was visualization of landscape-level data, and any development unrelated to this core functionality detracted from visualization development. However, this participant did not feel it was appropriate to voice this opinion because they believed it went against the expressed needs of the group. Since co-production is a collaborative effort, participants within the collaboration that are most vocal may have more weight in the direction of the co-production process, resulting in a situation where the “squeaky wheel gets the grease” (Environmental Team 1 member C, 2018 interview). While such dynamics are inherent in group activities, in the instance of co-production, the output or knowledge produced may not be useable and useful to all participants even if they had been a part of the process.

5.3.4 Relevance of co-production and changes in practice

While many of the participants in the VISTAS project already had some experience working in collaborative settings, all reported impacts from their group participation in the VISTAS project. Some of the impacts were related to how they would approach future work. For example, participants described how they would seek out work on interdisciplinary teams that involved group collaborations since they believed that these work environments resulted in getting a “a lot more done when you have a bunch of experts who have completely different backgrounds” (Environmental Team 1 member B, 2018 interview). For other participants who already had substantial experience with group collaboration and interdisciplinary work, the VISTAS project only solidified this
perspective, even though such collaboration were also viewed as higher risk. Participants who left the project and moved onto to new positions described taking the lessons from the VISTAS project with them, which, in many cases, was related to communication more generally. For example, participants in the VISTAS project were “forced to explain what [they] are doing” and the practice of explaining technical work and its limitations to non-scientists was a valuable skill (Environmental Team 1 member B, 2018 interview).

In this respect, changes in practice associated with co-production activities appear to be related to how people interact in a group and communicate with others, and, in particular how they interact with non-domain scientists. As one participant had noted “I think the magic happened when all these people came together...there is a chance to integrate all of the people who normally may not have had a chance to have a conversation,” feeling that “the actual process of them doing it [co-production] was more important than the actual output of the tool...talking through how to build something together is more important than what is actually built” (Domain Expert F, 2018 interview). While the team gained technical skills and advanced their scientific understanding of their domain, it was the communication and collaboration that was emphasized when discussing their experiences on the VISTAS project. It was this sentiment about the co-production process overall that led participants to be “very open to [co-production] after working with the VISTAS team” even if it presented a professional risk due to time constraints and less certainty in traditional measurable outputs.
5.4 Policy and Practice

5.4.1 Software design for domain-specific problems

The VISTAS project is an on-going effort between software developers, computer scientists, social scientists, environmental scientists, and ecologists, spanning eight years. By their own admission, participants observed that this was an unusually long time period for such a collaboration to exist. The reason for this long-term collaboration were many, including the expertise of those involved in the project in bringing together a diverse team of experts, as well as consistently providing the resources that allowed members of the Development Team to remain active. In the assessment of the participants in this project, this co-production process was successful in creating useful and useable software that met the domain needs of scientists. However, this was a resource and personnel intensive process that is likely difficult to replicate within other scientific domains and contexts. Fortunately, this does not necessary mean that the only solution to a deficiency in available software is to develop new software. Co-production can still be a viable method for addressing a variety of intractable problems related to software, application, and tool development, even across shorter time horizons. One way to apply such a software co-production approach in resource and time constrained contexts is to reduce technical inputs required to develop software.

Our work on the VISTAS project suggests that some of the features that we implemented could also have been implemented within more widely distributed software platforms with some modifications. This suggests an increased role for software systems to accommodate modular feature development, allowing for developers to focus on specific feature development rather than maintaining an entire software system. A
scripting window that could accommodate user-written programs was a feature discussed in the VISTAS project, and while it was not implemented, could be one approach for introducing a more modular framework to existing software. Some of the difficulty that members of Environmental Team 1 had with applying more widely available software to their modeling outputs had to do with both the specificity of their data, and challenges associated with selecting the appropriate routines for processing and visualizing this data. The customization or curation of generalized software platforms to fit the specific needs of scientific-domain users could address some of the software visualization challenges.

A movement toward more modular features and customizable interfaces for widely used software platforms could also facilitate software development that does not need to be completed by software engineers or programmers, but perhaps by team members with knowledge of computer programming. In our research, we found that some members of Environmental Team 1 had a high level of technical programming proficiency and would likely have the capacity to contribute to the development of such feature-based modules. It was the development and maintenance of standalone software platforms that was identified as too resource intensive for members of this team. Such a movement toward modular software, with more customizability and curation options, follows recent trends in computing, such as the R project which utilizes easy to install, user-written packages for statistical analysis. However, to the knowledge of our research group, such a software package does not exist for visualization.

5.4.2 Software development process

The VISTAS software design process revealed a few critical lessons about the creation of domain-specific software, and, in particular, the design of software for
visualization. As previously discussed, our participants began the co-production process with the desire to apply visualization techniques to their data, recognizing that they did not have the capacity to create such visualizations within their team. In the case of our project, this desire for visualization stemmed from the scientific understanding that topography was important for understanding landscape-level processes. For software developers facing different domain-specific requirements for scientists and/or non-scientists users, such desires could be quite different, and may not involve a data visualization component at all. Perhaps most importantly, collaborators may begin the process believing that visualization will help them understand their problem or research area and after iterations with developers, they may collectively decide that they instead require some other feature or design specification. In this respect, flexibility in the development and design process is imperative, and an emphasis of the co-production of knowledge literature more generally (e.g., Djenontin and Meadow 2018). How to build flexibility into the design process is a challenge since most teams have members with specializations—adding or subtracting team members may not always be an option given budgetary and other institutional constraints. To the extent possible, using networks to elicit expertise from people outside the project, especially with knowledge in areas that are new potential directions for development, could be a viable way to gauge whether this development approach is feasible or advisable, and if so, how to go about assembling the appropriate personnel to complete this work.

In the area of visualization software, one extension to the nested model for visualization design and validation (Munzer 2009; 2014) is to expand the “domain problem characterization” level to include this flexibility. Rather than the threat to
validity at this level being the “wrong problem,” using the language common in co-
production—“is solving this domain problem useful?”—could be more appropriate in
software contexts with applications towards complex or wicked problems. A similar
modification could be made to a user-centered design development approach (Brhel,
2015) where a criteria for contextual inquiry, as well as subsequent design iterations, are
crafted to specifically address the usefulness of features for addressing complex or
wicked problems and be flexible to such problems being redefined throughout out the
project.

5.4.3 Facilitating software co-production at institutional levels

The development period described in this research evolved across half a year, and
while it took considerable expertise, inputs, and trust to generate the conditions for this
co-production to occur, more explicitly directed co-production activities at institutional
levels could help lower the barriers typically associated with these interactions. For
example, when an agency is considering funding the development of software or web
applications for scientists and the general public, requiring this development process to
engage developers with intended users in a co-production process could help improve the
usability and usefulness of such an application. While such approaches may initially be
costlier than a developer creating a tool and then assuming that it is useable and useful for
an audience, the outcomes from such a process may be a more widely adopted tool that
effectively serves the public and others who interface with policy decisionmakers.
Requiring some dimension of co-production of knowledge, stakeholder involvement, or
other collaborative approaches as a criteria for funding is a direction that governmental
agencies such as the NOAA regional integrated science assessment (RISA) have already
encouraged (Meadow et al. 2015; Stevenson et al. 2016). Other government and non-governmental institutions providing funding for research programs with software development components should follow suit, especially for policy domains where complex or wicked-type problems are prevalent.

Another consideration for institutions that support such collaborative software development projects is not only to provide funding for creating software, but also providing funding that ensures its sustainability. Software maintenance is not a problem that is unique to co-produced software; however, unlike software developed using a more traditional approach, co-produced software is an extension of a community of participants, purpose-built to address specific domain needs, often of high societal importance to policymakers. Software such as VISTAS, developed for domain scientists, is likely to never reach the level of users or adoption compared to more generalizable software; however, this does not mean that its continued maintenance is without merit. Such software was co-produced because generalized software platforms did not meet domain needs, and when such domain-specific software is no longer maintained it is uncertain whether domain experts can find a replacement or substitute with the same set of features to complete similar tasks. If we accept the importance of co-producing software to address domain specific problems that are complex, wicked, or of otherwise high societal importance, then a plan to make available the product of these efforts for reasonable time periods after the conclusion of projects must be established. Whether this becomes the responsibility of the funding institution or software developers, it should be a priority to avoid creating a graveyard of nonoperational domain specific software.
5.4.4 Co-production as an approach for knowledge creation

When we consider the lessons from this co-production effort alongside a collection of other co-production efforts described in the literature, some patterns for best practices and policies emerge. First, our co-production effort was a time and resource intensive process that thrived in part because of the dedication of project participants and their willingness to be involved in a process with more uncertain outcomes than traditional research approaches. While these factors certainly contributed to the longevity and success of VISTAS, reliance on such factors for successful co-production limits the application of co-production to certain problem contexts, stakeholders, and inputs. For example, in wicked problem contexts with both high uncertainty and high societal importance, recruiting stakeholders willing to participate in an uncertain knowledge production process may cognitively create too many layers of uncertainty that discourages participation. One recommendation, therefore, is to reduce some of the uncertainty associated with the co-production process by emphasizing and rewarding outputs that are not conventional deliverables. One possible way to achieve this is through governmental organizations and academic institutions formally recognizing co-production contributions as comparable to other forms of productive outputs, incentivizing participation.

Second, co-production approaches are often applied to intractable or wicked problem contexts, which, by their very nature, have no easy solution. When co-production efforts attempting to address such wicked problems fail or do not achieve the endpoints established by participants, this may give the impression that co-production is ineffective, or that co-production efforts are not as reliable in producing knowledge
compared to other, more traditional approaches. A better comparison of co-production to other traditional approaches would require both approaches to be applied to the same problem context. Examples of this are difficult to evaluate because co-production is often a context-specific process undertaken after other approaches have either failed or deemed a poor fit. In this sense, co-production applications may be more akin to a critical care unit of a hospital: the unit’s death rate is higher because it treats very sick patients. When institutions are evaluating the effectiveness of co-production efforts or making decisions about funding co-production projects, the inherent challenge of addressing these wicked problems and the relative effectiveness of suitable alternative approaches should be considered.
6 Conclusion

The VISTAS project is a collaborative and immersive research effort that brought together experts from a variety of disciplines including computer science, software engineering, ecology, environmental science, and social science. A result of co-production among this diverse group of stakeholders was new features developed for VISTAS software, a data visualization platform designed to display complex topographic information at landscape scales. New software features were well-received by both developers and scientist stakeholders, with many participants commenting that the process of collaboratively producing software was as impactful to their science and research approach as the newly developed software features that were purpose-built to meet their domain specific needs. Critical components of this research, and lessons learned from this effort, are described below.

In this research, I first identified domain-specific challenges associated with visualizing environmental and ecological data, as well as established the importance of visualization for environmental scientists to understand and explore their modeled datasets. For our participants, much of the phenomena they studied evolved across spatial and temporal scales. The VISTAS project participants were particularly interested in being able to integrate statistics and data analytics with landscape-level visualizations which drove the most recent cycle of new VISTAS software development that was the focus of this study.

In this software development period, the Development Team added features to the existing VISTAS software platform through an iterative, co-production process involving continual input from all members of the VISTAS team. Some important factors were
identified for facilitating this co-production effort. First, there were dedicated team members who had been involved in the process since its inception and were critical in both generating a constant source of funding as well as bringing together a diverse group of participants and facilitating their interactions. Second, due to the experience of the personnel involved in the product, there was a feeling of trust and inclusivity that motivated people to engage in the co-production process. Third, the project evolved across a period of time that was identified as atypical for co-production efforts, spanning eight years and involving many participants, some of who were involved for the entire length of the project, and others for much shorter time periods. This allowed project objectives to grow and adapt to changing participant need and meet new scientific and technological demands. The result of this process was co-produced visualization software that was perceived as both useable and useful for collaborators.

While this project was considered a success by participants, there are limitations to the conclusions that can be drawn from the experiences of our project participants, domain-specific software co-production, and co-production and software development more generally. Some of these limitations are more broadly related to the case study approach that was applied, while other limitations are specific to the context of this research. A summary of these limitations is described below.

This research describes the co-production of a specific software platform, in a domain-specific context, for a single use case. Since this case study examines the co-production of visualization software for environmental scientists and ecologists, applications of this study’s findings are constrained to this research context. For example, one way that the generalizability of this case study is limited is through its comparability
to other cases and contexts: there is no indication for how similar or different this experience of co-producing software is to other co-production or software development contexts. How unique this study context is to the domain-specific problem that was being addressed is challenging to ascertain, but feedback from participants who were involved with other software development projects tended to describe the development process as unusual with respect to their previous experiences. While generalizability was never a goal of this case study research, or even achievable using this design, increasing the number of users to include other environmental research teams could have helped broaden the study’s scope. Indeed, we had tried to recruit an additional environmental research team but ultimately only one of these teams participated in the co-development process. Having one environmental research team participate in the co-development process not only limited the number of participant-users, but also reduced the variation in the way that the software was being applied to domain-specific environmental and ecological problems and participant reactions to its usefulness and usability.

Other limitations to this study are related its set up and design. For this research, we considered a development cycle that was analyzed as a single iteration of VISTAS software development. The timing of this iteration was influenced by factors that included funding, involvement and attention of scientist-participants, and the availability of critical software development personnel. Additionally, when the pre-period interviews began, and when the post-period interviews ended, was organized in a way that fit my own academic schedule. In this respect, the research design of this study, while containing elements that were systematic, also had features that were selected through convenience.
This research considers a year-and-half long segment of a project that spanned eight years. This recent segment includes a development cycle for a comparatively mature VISTAS product where many of the initial visualization needs of participants had already been met. If similar pre-post research was conducted for a development iteration in an earlier phase of the project, such as when the product required critical bug fixes or lacked import tools for certain data formats, findings related to the usefulness and usability of the software product would have likely been substantially different. So, while conclusions drawn at the end of the research were based on the most recent development iteration of VISTAS, such conclusions may not be applicable to earlier phases of the research or describe the experiences of participants involved with the project at other iterations.

There are also limitations associated with my own knowledge of computer software design, having a background primarily in social science research. While I gained some exposure to the process of developing software, there were likely many areas where I did not have the knowledge base required to fully grasp more intricate and nuanced aspects of the software development process. While this lack of computer knowledge was at times useful because it encouraged developers to distill complex concepts into simpler terms, important information was likely lost in this process. Similarly, I do not have a knowledge base in the domain-specific areas of our scientist-participants, including environmental science or ecology. This lack of knowledge could have limited my ability to understand information and communicate with scientist collaborators, influencing the way I interpreted and articulated the challenges, needs, and feedback from these domain scientists. While it is difficult to gauge to what degree my lack of knowledge in software
development and environmental science and ecology may have impacted this study, one plausible outcome is a more narrow presentation of technical and domain-specific information throughout this research.

In this research, I considered an indirect pathway for which the process of co-production influences the public policy process. Such a pathway enabled ecologists and environmental scientists, who interfaced with stakeholders and other cross-sections of the public, to develop knowledge to be applied toward informing land management policies. Another, more direct route, would be to embark on a visualization co-production exercise directly with these stakeholders and public. Such a co-production effort would likely involve expertise and experiences from computer scientists and programmers, environmental scientists and ecologists, and members of the public that engage in management decision-making. The intention of such a co-production effort would produce software that is both useable and useful to a broad background of users, perhaps moving towards visualizations systems that are web-based rather software installed on a personal computer.

There are additional future directions to extend this case study research. One possible area for further research could consider assessing how other environmental scientists and ecologists who were not part of the VISTAS project gauge the usefulness and usability of the VISTAS software for their research applications. This would help explain whether the needs of our participants for displaying modeled data topographically is specific to our stakeholders, or more general to the requirements of other scientists in this domain. Another direction for future research would be to use a similar co-production approach applied in the VISTAS project to develop visualization software in a
different scientific domain and compare the development process and project outputs with the findings in this study. Yet another area for future research could consider co-production of software unrelated to visualization yet applied toward some other problem context in environmental science and ecology, such as the development of a software system for managing and sharing big data. These are all exciting areas for future research and would help further elucidate themes identified in this study such as the necessity of domain-specific software, the effectiveness of co-production in designing software for addressing wicked problem contexts, and the role of visualization in communication and the scientific process.
7 Bibliography


[https://doi.org/10.1016/j.ins.2014.01.015](https://doi.org/10.1016/j.ins.2014.01.015)

[https://doi.org/10.2307/41703503](https://doi.org/10.2307/41703503)

[https://doi.org/10.1109/MCG.2013.39](https://doi.org/10.1109/MCG.2013.39)

[https://doi.org/10.1016/j.jvlc.2007.02.006](https://doi.org/10.1016/j.jvlc.2007.02.006)


[https://doi.org/10.1057/jit.2014.17](https://doi.org/10.1057/jit.2014.17)


Frantzeskaki, N., & Kabisch, N. (2016). Designing a knowledge co-production operating space for urban environmental governance—Lessons from Rotterdam,
https://doi.org/10.1016/j.envsci.2016.01.010


https://doi.org/10.1890/120103


https://doi.org/10.1177/0162243905283639


https://doi.org/10.1016/j.respol.2008.01.008


https://doi.org/10.1080/10919392.2015.1125187


Winters, K. M. (2015). File | *WintersKirstenM2015.pdf* | ID: g158bk82w | *ScholarsArchive@OSU*. Retrieved from [https://ir.library.oregonstate.edu/concern/parent/47429c578/file_sets/g158bk82w](https://ir.library.oregonstate.edu/concern/parent/47429c578/file_sets/g158bk82w)


8 Appendices

8.1 Appendix A: Pre-development interview protocol

Baseline Phase Protocol Script

Introduction: As you know, we’re working with scientists like you who produce large amounts of data and want to find ways to visualize that data for both analysis and communication of results. Before we get started on that effort, we’d like to ask you a few questions about your data and what you expect from participating in this project. What questions do you have before we get started?

1. Please tell us about the data you collect and would like to visualize in this project.
   - What kind(s) of data do you want to visualize?
   - At what scale(s) do you work?
   - What does the output of your data look like?
   - What are some data ‘challenges?’ (Too big? Too varied? Too fast/streaming?)
   - How do you currently analyze the data?
   - What kinds of visualization have you used in the past to display data for either analysis or communication?
   - What difficulties have you had with data analysis or display of this data?

2. What do you think about your current abilities to produce visualizations?
   - Do you have the tools to do what you want/need to do?
   - Do the tools have all the options you want/need?
- Have you modified tools (software, etc.) to make them work for you?

3. Please tell us what you expect to gain from visualizing your data?
   - How it will help with data analysis?
   - How it will help with communicating results?
   - What problems do you expect while creating visualizations of your data?
   - How could we mitigate those difficulties?

4. Please tell us what kinds of data visualizations will be the most helpful to you for your data analysis?
   - What about visualizations for communicating results to other scientists? Non-scientists?
     What do you think is most difficult for people encountering visualizations for the first time? How can we mitigate those difficulties?

5. What questions do you have for me? About the project?
8.2 Appendix B: Post-development interview protocol

Post-Development Phase Protocol Script

Introduction: You’ve completed a set of visualizations with the VISTAS team and we’d like to talk with you about the whole process this time, looking back from your current position and experience. What questions do you have before we get started?

1. Please tell us about ways that the visualization process helped you think about your data?
   - How did it change your understanding of ecological processes?

2. Please tell us about ways that the visualization(s) helped you communicate with others about your findings?
   - Other scientists, funders, specialists?
   - Non-scientists, stakeholders, others?

3. What did you learn on this project that you will take with you into future research efforts?
   - about data collection?
   - About data analysis?
   - About visualizations?
   - About communicating results?
4. If you could start your project over from the very beginning knowing what you know now, what would you change?

5. What advice do you have for computer scientists working with non-computer scientists on visualization projects like this one?

6. What questions do you have for me? The project?