**Recreational Fisheries Decision Support Tool: A shiny app for Fisheries Managers**

Kimberly Bastille

Andrew Carr-Harris

Scott Steinback

May 17, 2024

## Introduction

Recreational fisheries managers are challenged with making decisions that will impact millions of individuals and their communities (cite). To make informed decisions, managers work with scientists whose mission it is to provide the scientific information and tools necessary for productive, sustainable, and healthy marine ecosystems and coastal communities (cite). This collaboration is an integral part of the Magnuson-Stevens Fishery Conservation and Management Act (cite) and can be difficult for many reasons including the gap in technical and non-technical knowledge between the scientists and managers and the amount of time it takes to compute the advanced mathematics that go into the science shared. These are challenges faced in the Summer Flounder (Paralichthys dentatus), Black Sea Bass (Centropristis striata), and Scup (Stenotomus chrysops) recreational fishery in the U.S. Mid-Atlantic region. In this paper, we will discuss how we were able to overcome these barriers by building and disseminating a recreational fisheries decision support tool R shiny application for fisheries managers by following the software development life cycle and focusing on collaboration

R shiny applications have become increasingly popular since its initial roll out in 2016 (Kasprzak et al. (2021)). In that time, many have been developed for fisheries purposes including reference point exploration (Huynh et al. (2023)), reporting (Noleto-Filho et al. (2021), cite), and monitoring ((**Tilley2020?**)). The recreational fisheries decision support tool shiny application is the first to be used directly in the management regulation setting process. This is possible because of the management strategy used in this fishery.

In 2022, the Mid-Atlantic Fishery Management Council and the Atlantic States Marine Fisheries Commission adopted the “Percent Change Approach” for setting recreational regulations (size limit, possession limit, and open fishing season) (cite). This approach looks at two factors when determining when a change in management regulations should remain status-quo, be liberalized, or be restricted - 1) Confidence interval around expected harvest under status-quo regulations and 2) biomass compared to target. Together, these determine the percent change in harvest estimate for the following two year management period (cite). The model used to determine the estimate of harvest is a recreation demand model which explores how changes in size limit, possession limit, and open fishing changes impact estimates of harvest, discard, effort, and economic value of the fishery (tech memo).

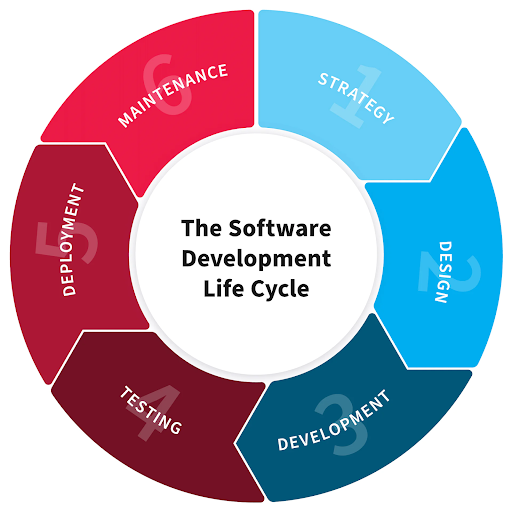
## Model Background

The Decision Support Tool uses the recreation demand model to calculate the outputs required by managers to meet the percent change required by the above approach. It incorporates catch-per-trip data from the Marine Recreational Information Program (cite), length distributions from stock assessments (cite), and angler behavior from a discrete choice experiment (tech memo). The outputs provide estimates of harvest and discards (as well as dead discards) in both pounds and numbers of fish, estimates of effort in number of trips, and an estimate of angler satisfaction in U.S. dollars. The model is broken into three fishing modes (shore, private boat and for-hire boat fishing) (tech memo).

In 2023, we developed an R shiny application which provides managers with a user-interface to interact directly with the model. The development of this application has fundamentally changed the workflow for determining recreational fishing regulations for these species. The previous method required an individual model developer to run the model over 200 times on their local machine. Managers would email the modeler requesting different regulation alternatives and that individual would then email the results in a day or two. This new method allows the manager to run the model with their regulation suggestions and receive output in about 15 minutes. Thus, managers are able to explore more regulation alternatives and compare outputs on their own schedule.

## Software Developement Life Cycle

Viewing this Decision Support Tool Application not only as a product but as an iterative workflow allows us to apply the traditional software development framework. There are many ways to visualize Software Development Life Cycle models ((**Shylesh?**) 2017). For this project we focus on the circular workflow (Figure 1) due to the iterative nature of fisheries management. The framework consists of six phases (Strategy, Design, Development, Testing, Deployment, and Maintenance). In each phase of the framework we engaged the recreational fisheries managers for whom the tool was built. We believe this collaboration helped us build a useful and intuitive product that the managers were excited to implement.



Software development life cycle visualization describing the workflow applied to build and maintain the Recreational Fisheries Decision Support Tool Shiny Application. (<https://nix-united.com/blog/software-development-life-cycle-nix-approach-to-sdlc/>)

### Strategy

The first step is to understand the problem, understand the requirements of that problem, and to plan ((**Mishra2023?**), Shylesh (2017)). The goal of this project was to build a product that allowed summer flounder, black sea bass, and scup recreational fisheries managers in the U.S. Mid-Atlantic region to interact with a recreation demand model used in their management process. Managers needed the ability to explore how changes in bag limit, possession limit, and open fishing season impact estimates of harvest, discards, effort, and the economic value of the fishery. They required that the user-interface be accessible online and be able to run in a reasonable amount of time.

Our strategy to meet these needs included 1) to develop a shiny application, 2) to deploy that application in the cloud and 3) to engage managers throughout the project to both update them on the progress and to seek feedback.

### Design

Once we had a strategy, we began planning the execution. In this scenario with relatively few developers (1 app and 1 modeler) and users (~30), we were able to stand up a working group (add website??) composed of a mix of individuals interested in Summer Flounder, Black Sea Bass, and Scup Recreational Management in the Mid-Atlantic region of the U.S. The folks included state and federal managers, stock assessment scientists, …

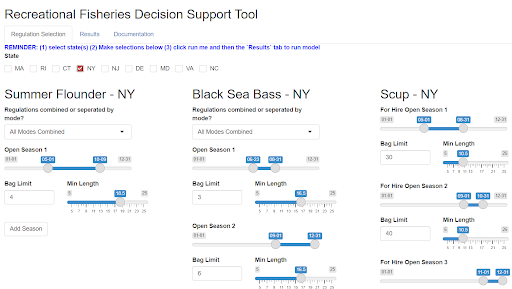
Over the six months it took to develop the application and make it available to users, this group held monthly meetings to facilitate communication about the product. Through this engagement, it became clear the major attribute users wanted to see was flexibility. To support this we included the ability to split management by mode. States who previously managed shore, private boat, for-hire boat fishing as a single unit are able to click a button and options to manage each mode separately pop down. Furthermore we also included the option to add a fishing season so managers were not limited to the number of fishing seasons they had before. Co-designing this tool with the users allowed for a product that the users were invested in and something we as the developers knew was going to meet their needs.

### Development

The Decision Support Tool Shiny Application was developed in R shiny, an R package that allows programmers to build web applications (Chang et al. (2022)). This interactive application allows non-technical users to interact with R code and thus, run the recreation demand model. Development co-occurred with the design input from the users of the tool. Through this they were able to see regular progress updates on the application and see how we as developers were incorporating their feedback each month.

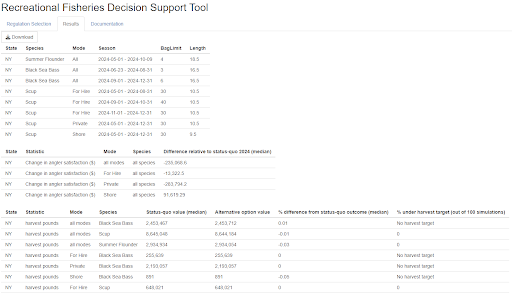
Throughout the application development we were also making improvements to the model as well. Adding features to the model that would enhance the capabilities of the tool as well as capturing more uncertainty inherent in the underlying data (tech memo). These improvements were also communicated to the users of the tool. During model-focused meetings we sought their direct input on data decisions and sought feedback on the model. We also presented a detailed overview of the model so they would not see the application as a black box but as a front-end for the model they understand.

The shiny application contains three components. The first is the “Regulation Selection” tab ((**RegulationSelection?**)) in which the user selects which state(s) they wish to explore outcomes. When a state is checked, the default management regulations drop down. Defaults are the previous year’s management regulations. The user then has the power to change the open fishing season (when people can legally keep fish), possession limit (the number of fish an individual angler can keep per day), and/or size limit (the minimum size of fish and angler can keep) for the species that have been determined for restrictions or liberalization based on the Percent Change Approach.



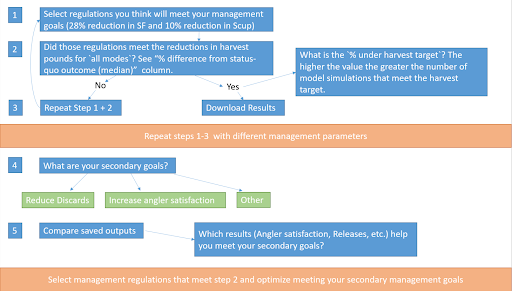
Recreational Fisheries Decision Support Tool Shiny Application “Regulation Selection” tab showing New York selected.

The second component is the “Results” tab ((**ResultsTab?**)). Here, the user can see how the regulations they selected on the previous tab impact estimates of harvest, discards, fishing effort, and angler satisfaction. These outputs are presented in five tables and can be downloaded for later review and comparison between alternatives. The first table shows the regulations that the user selected on the previous tab. The second shows change in angler satisfaction, measurement of the compensating variation (CV) generated from the change in fishing trip outcomes in U.S. dollars (tech memo). The harvest table is the most used and shows the status-quo, alternative, percent difference between status-quo and alternative, and the percent of model runs under the harvest target. These variables are presented for each state select and for all modes. The next table shows the same variables excluding percent under target for the estimated discards and dead discards. The final output is an estimate of fishing effort measured by the number of trips expected under the alternative regulations.



Recreational Fisheries Decision Support Tool Shiny Application “Results” tab showing top tables of model outputs for NY.

The percent difference between status-quo and alternative is the measure used in the Percent Change Approach and is thus the most important variable for this application. The other values are offered as supplemental information to help managers make the decisions that are best for their anglers. This being the first year that managers would see the outputs in this format and use them to inform their management decisions, we developed a decision tree to help them use the non-harvest variables provided ((**DecisionTree?**)). Together with managers, we discussed how they would use the tool and how we as developers imaged them interpreting and applying the results.



Recreational fisheries decision support tool decision tree: a framework for helping managers use the tool to inform their decision making process.

The decision tree consists of five steps, mostly questions the managers can ask themselves as they use the decision support tool. The primary goal for managers is to meet the liberalizations or reductions prescribed by the Percent Change Approach (28% reduction is summer flounder and 10% reduction for scup in 2024). The secondary goal will vary by the interest of the anglers in each state. One state may want to optimize angler satisfaction while another may want to focus on reducing discards. For both scenarios, the managers run multiple iterations of the model with varying alternative regulations that they believe will meet their primary goals. Once they have 3-4 alternative management measures that meet the requirements of the Percent Change Approach, they can compare results to meet their secondary goals, whatever those may be. This strategy allows them to bring multiple options to their anglers and facilitates open communication and understanding around the regulation setting process.

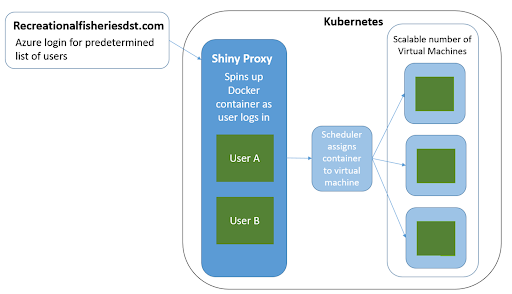
### Testing

The application was tested in multiple stages throughout the development and deployment phases. A beta version of the application with limited capabilities (one state) was made available for users to explore and test as soon as the application was deployed in the cloud. At this point we were still building out the application to include the other eight states but we wanted the users to get a feel for how to use the tool before their management decisions were due.

In the grander scale, this initial iteration was a test for the deployment in the cloud. We did not know exactly how much usage the application would accrue during this process. The findings from this first year will be used to inform further budgeting decisions as well as the strategy and design phases for related projects, specifically the build of a similar decision support tool for the Gulf of Maine Cod and Haddock recreational fishery.

### Deployment

To support the shiny application and address the user’s concern of how much time it takes for an individual to run the model and collate the results, we deployed the shiny application using Microsoft Azure Cloud Services (Azure). This allowed us to put the tool in the user’s hand and thus alleviated the burden of computing the information that managers need in their decision making process from a single individual scientist. The workflow, developed by Microsoft cloud engineers ((**backend?**)), allows us to meet the needs of both the model (processing power) and the users (a public-facing website where they can access the tool).



Azure deployment visualization outlining the backend workflow developed by Azure Cloud Engineers.

Multiple software tools were used to meet these needs within Azure. The process is containerized using Docker (Merkel (2014)) allowing us to package our data and code and port those to any machine. Having the code in a container means we can spin up as many instances of the model as there are active users meaning each user is working in their own container.

First ShinyProxy deploys and manages the shiny application (<https://www.shinyproxy.io/>). To do this, it spins up a docker container which contains the data pathways, code, and configuration necessary to run the shiny application. Once the container exists Kubernetes, a container management system (<https://kubernetes.io/>), uses a scheduler to assign the container to a virtual machine. This process happens each time a user logs in through the Azure login portal at recreationalfisheriesdst.com. Meaning each individual user is running the shiny application in their own workspace and thus, not competing or queuing for runtime. Each user is interacting with the model in their own workspace.

The number of virtual machines are scalable meaning as usage of the tool increases more machines can be created and when users log out or time out of their session the virtual machine they were working on can be shut down. This helps manage the cost of the Azure services as we needed to drastically increase the size of the virtual machines we were using. Increasing the size of the machines from 32G and 16 cores to 756G and 36 cores allowed us to parallel process (“futures” R package) the 100 iterations of the underlying model instead of sequentially processing in a for loop as was previously done. This significantly reduced the amount of time to run a single model from about 6 hours to about 15 minutes per state. Fifteen minutes is much more reasonable for the users though future reduction in runtime was requested.

### Maintenance

#### Open Science

It is well documented that embracing open science strategies and tools results in increased efficiency, transparency, and repeatability in iterative workflows ((**Bastille2020?**), (**Lowndes2017?**)). Applying those same principles here ensures that 1) we can more efficiently work with our current and future selves by setting up protocols for storing and documenting data and code and 2) users of the tool have access to the backend preventing potential contentions with lack of transparency (github link?).

Storing code developed on a local machine on github means that the app developer is able to make quick changes to the application and pick up those changes in the Azure workspace. This addresses (1) by pulling the updated code version in Azure and rebuilding the docker image used to build the container. Both minor and major changes to the shiny app can be made available to users in minutes by using this workflow. This allows us to maintain the application in the short and long term.

#### User Survey

In preparing for the next cycle, we surveyed the users to gather feedback on how they would like to see the application and their user experience improve for next year. We will be using this list of ideas to prioritize what features to include in the strategy and design phases of the next lap around the software development life cycle. A major take away from this survey is that the app “was intuitive to use” and that users felt ownership of the product and invested in its continued improvement. Moving forward we will continue our collaboration strategy by engaging the users both in formal meetings, more casual office hours, and in written feedback. This allows users to interact with developers in the format they are most comfortable using meaning we are soliciting feedback from everyone and not just the loudest in the room.

## Conclusion

Applying the software development life cycle for a recreational fisheries management decision support tool allowed us to follow a systematic framework for meeting the needs of our users and our tool. The six steps outlined above are not one-off practices and the order could vary slightly as we iteratively improve the product. There are three major takeaways from this workflow. First to engage the stakeholders, in this case the recreational fisheries managers, early and often. This builds trust and ensures that the final product is adapted to individual needs. We engaged the managers in each step of the framework and because of this we were able to deliver a product that was used directly in management regulation setting and that they were excited to use.

Second, is to embrace technology. Delivering this tool to managers in six months would not have been possible without deploying the shiny application in Azure. This tool needed to overcome two main barriers to usage (1) the runtime and (2) it needed to be public facing and accessible to users. Cloud services solved both of these challenges. By combining R’s parallel processing capabilities and the larger virtual machines available in the cloud space we were able to dramatically reduce the runtime. Through Azure cloud service we were also able to purchase a website domain (recreationalfisheriesdst.com) and deploy the application there making the tool available online to those with the link and a login.

Lastly is to ensure transparency. Transparency in all forms means showing our users what is behind the model reinforcing the first major takeaway, openly discussing the capabilities and limitations of the tool, and using open science strategies and softwares to ease future work. Tools like Git enable the application developer to quickly work with themselves on different machines to update the tool by passing the code through the version control software. This makes the code available to those interested in the math behind the model, and allows model and application developers to collaborate with each other more efficiently. Transparency in each phase of the framework leads to more efficient work and greater trust in the product developed.

Moving forward we expect this workflow to change and adapt as we continue to collaborate with recreational fisheries managers to develop a decision support tool that can meet their needs and inform their decision making process. We plan to use their feedback and the lessons learned from this first cycle to make improvements to the model, application, and deployment of the tool. We will continue to apply the six phases of the software management life cycle (Strategy, Design, Development, Testing, Deployment, and Maintenance) for this application and will follow that same practices and apply the lessons learned as we expand for Gulf of Maine recreational Cod and Haddock which are managed using a similar model by the New England Fisheries Management Council.

## References

Chang, W., J. Cheng, J. Allaire, C. Sievert, B. Schloerke, Y. Xie, J. Allen, J. McPherson, A. Dipert, and B. Borges. 2022. [Shiny: Web application framework for r](https://CRAN.R-project.org/package=shiny).

Huynh, Q. C., T. R. Carruthers, and A. R. Hordyk. 2023. [Reference point calculator (RPC): A shiny app for exploration of limit reference points for fisheries management](https://doi.org/10.21105/joss.05418). Journal of Open Source Software 8(86):5418.

Kasprzak, P., L. Mitchell, O. Kravchuk, and A. Timmins. 2021. [Six years of shiny in research - collaborative development of web tools in r](https://doi.org/10.32614/RJ-2021-004). The R Journal 12:20–42.

Merkel, D. 2014. Docker: Lightweight linux containers for consistent development and deployment. Linux journal 2014(239):2.

Noleto-Filho, E. M., R. Angelini, J. Steenbeek, and A. R. Carvalho. 2021. [New, flexible and open-source fisheries self-reporting app: The Shiny4SelfReport](https://doi.org/10.1016/j.softx.2021.100843). SoftwareX 16:100843.

Shylesh, S. 2017. [A study of software development life cycle process models](http://dx.doi.org/10.2139/ssrn.2988291). SSRN.