Implementing *Dynamic Recovery Plans*: Evaluation and Recommendations

Defenders of Wildlife, Center for Conservation Innovation

The current standard of static, "paper"-based Endangered Species Act (ESA) recovery plans is a significant barrier to improving the recovery of listed species. We developed four types of dynamic recovery plans and evaluated each according to nine criteria that address the needs of transitioning to a new model for recovery planning. Based on our evaluation and discussions with the Services, we recommend that the Fish and Wildlife Service and the National Marine Fisheries Service (collectively, Services) use the WordPress implementation as the core of dynamic recovery plans, supplemented by other platforms for data integration. Using WordPress meets the fundamental needs of modernizing ESA recovery planning, does so in an easy-to-use (and mobile-ready) way, and is cost-effective.

ESA recovery plans as currently implemented are snapshots of what is known about a species' biology; the threats it faces; its status; and what is needed to recover the species (actions, money), *at the time the plan is finalized*. Current recovery plans have a median age of 19 years, and most have never been revised to reflect new information. We know that our collective knowledge about ESA-listed species changes rapidly, so the static plan approach is flawed from the start. These static plans are not able to incorporate new data and ideas about the kinds of information that can inform recovery. For example, web-based tools for real-time data updates to graphs and tables of, e.g., regulatory permitting, can be rapidly deployed on the Internet, but not in static plans. And static plans are inherently non-interactive, which means that conservation partners, academics, and other interested parties are hampered in their ability to participate in recovery planning.

Methods

To address the shortcomings of static recovery plans, we developed five different dynamic recovery plans (bold links point to the instance of each plan):

1. ***Shiny application*** *(fully custom):* Using the Shiny framework for the R programming language, we developed a web-based dynamic recovery plan that directly integrates data import, analysis, and graphing in a single application. Development at this point is only on the front end, but features like security, collaborative editing, and PDF rendering can be developed.
2. ***GitBook****:* Building directly off of the GitBook platform, we rendered a git- and markdown-based dynamic recovery plan.
3. ***Discourse****:* Discourse is dubbed the "Civilized Discussion Tool Kit" and was originally designed as a successor to classical internet forums. We re-purposed a Discourse instance for a dynamic recovery plan because so much of recovery is about having

discussions among different parties.

4. ***Google Docs****:* We created a dynamic recovery plan in Google Docs (see also GSuite). Although less flashy than the other options, the word processor base of Google Docs is a direct link to existing plans.
5. ***WordPress:*** WordPress may be the most widely used Content Management System (CMS) on the internet. Although mostly used with blogs, WordPress can be used to drive entire websites; in fact, defenders.org is built with WordPress!

Each of these four formats can - or can be developed to - meet nine criteria to varying degrees:

1. *Familiar user interface (UI):* A familiar UI is needed to facilitate adoption of dynamic recovery plans by personnel from the Services. The more dissimilar the UI from past experience, and the more new points that need to be learned as part of the user experience (UX), the less appealing a new technology will be. The starting point for most people will be a word processor like Word or Google Docs.
2. *Secure system:* Whatever technology is used to create and present dynamic recovery plans has to be secure. The Services need to retain control over the content of plans, which may mean authorizing other qualified parties to provide information. 1024 - 2048 bit encryption.
3. *Collaborative editing:* One of the main benefits of dynamic, web-based recovery plans is having the ability to do real-time or near-real-time collaborative editing.
4. *Community interaction:* Dynamic recovery plans should provide the capability for "the community," whoever they should be, to learn about what species need, to contribute ideas about recovery, and so-forth.
5. *Embedded data/graphs:* Static recovery plans might include data up to the time the plans are finalized, but will be out-of-date immediately thereafter. Dynamic plans need to be able to show or link directly to up-to-date graphs, maps, and tables that can help improve species recovery.
6. *Track all changes:* To ensure traceability of any changes to a recovery plan, every single change needs to be recorded.
7. *Print PDF:* There will be some cases in which printed or otherwise "fixed" copies of a recovery plan are needed. Because the Portable Document Format, PDF, is the industry standard for viewing documents across computer operating systems, dynamic recovery plans should be exportable as PDFs.
8. *Development cost:* The Services are under severe budget constraints, so developing a platform for dynamic recovery plans should be minimal: as much funding should go to implementing recovery actions as possible rather than developing the platform.
9. *Maintenance cost:* As with development cost, the cost of maintaining dynamic recovery plans should be minimized so that more recovery actions can be funded.

We attempted to be as objective as possible when considering each of the four dynamic plan types in each of the nine categories. We assigned scores (range 0 - 5), with an objective of maximizing each score component and maximizing the total score. While we could also

rank the different versions, we found the results were effectively identical.

Results

All five versions of dynamic recovery plans are linked above. The score-based table of results[1] is:

Criterion	Shiny-based**	GitBook	Discourse	GDocs	WordPress
Familiar user interface	1	3	2	5	5
Secure system	0	4	5	5	5
Collaborative editing	0	3	3	5	4
Community interaction	0	3	5	4	4
Embedded data/graphs	5	3	3	2	4
Track all changes	0	5	4	5	5
Print PDF	0	3	1	5	4
Development cost*	0	3	3	4	4
Maintenance cost*	0	3	3	5	4

Total Score	6	30	29	40	39
* low score = high cost					
** requires extensive development					

Shiny

There are two advantages to the Shiny-based recovery plans, one of which was not scored. First, we can embed data - as graphs, maps, or tables - directly into the Shiny plans with very little coding overhead. Second, and much more superficial, the Shiny plan has a distinctive look and "feel." But in the other eight scored criteria, Shiny apps have the lowest score or the highest rank, i.e., it is the least desirable choice. All nine criteria can be met, but doing so will require time and money that could be spent on recovery actions. Because all these things are possible, but not yet implemented, we limit further discussion.

GitBook

The GitBook format consistently scored (was ranked) reasonably well because, like the Discourse format, it meets most criteria out-of-the-box. Because it is git-based, GitBook was highest-rated for tracking changes. Note that the score for this criterion doesn't reflect that git has both strengths - it can track changes in arbitrarily complex change networks - and weaknesses - it can be very hard to learn because of the power. The collaborative editing abilities of GitBook are directly tied to tracking changes, which scores lower than GDocs because (a) authors can't see changes being made concurrently and (b) git can be really complicated (see above). The GitBook editor is pretty well-developed, but it lacks certain features that would need to be developed and incorporated in the editor, such as

[1] Note that both tables are linked directly to a Google Sheet and update automatically.

the ability to center or right-align images, generate the link-to-app buttons, etc. Even with such added buttons to simplify writing, GitBook editing is really different from “standard” word processor writing; in addition, the workflow using git is substantially different from a “standard” workflow. Last but not least, GitBooks can readily generate PDFs (or eBooks) of their contents, but the formatting - which can be adjusted with style sheets - will almost always be a bit wonky.

Discourse

We found the Discourse system most intriguing because it is the most different from standard recovery plans in many ways, but is familiar in others. While it scored and ranked very similar to GitBook, the strengths and weaknesses were in different criteria. Perhaps the best feature of Discourse is the community interaction: this idea is fundamental to the platform. Another great feature is security, from the strength of encryption (2048-bit keys) to the high granularity of access control. A weakness is that while all changes are tracked in the underlying database, a tool to extract those changes easily would need to be developed. And like GitBook, the Discourse editor is well-developed...but is also rather different from what most people will be used to.

Google/GSuite Docs (GDocs)

As noted in the executive summary, the GDocs solution isn’t flashy. A user from the Services, other agencies, conservation partners, or any other people wouldn’t necessarily recognize a GDoc-based solution as something unique on first glance. But with the exception of some stickiness with community interaction (there is some access control but nothing fine-grained) and indirect support for embedded data/graphs/maps (common to all except the Shiny solution), GDocs has the best scores and ranks among the other seven criteria. The UI is completely familiar; the workflow is completely familiar; security is taken care of; and so-forth. Aside from aesthetics, we found few arguments against using GDocs as the basis of dynamic recovery plans.

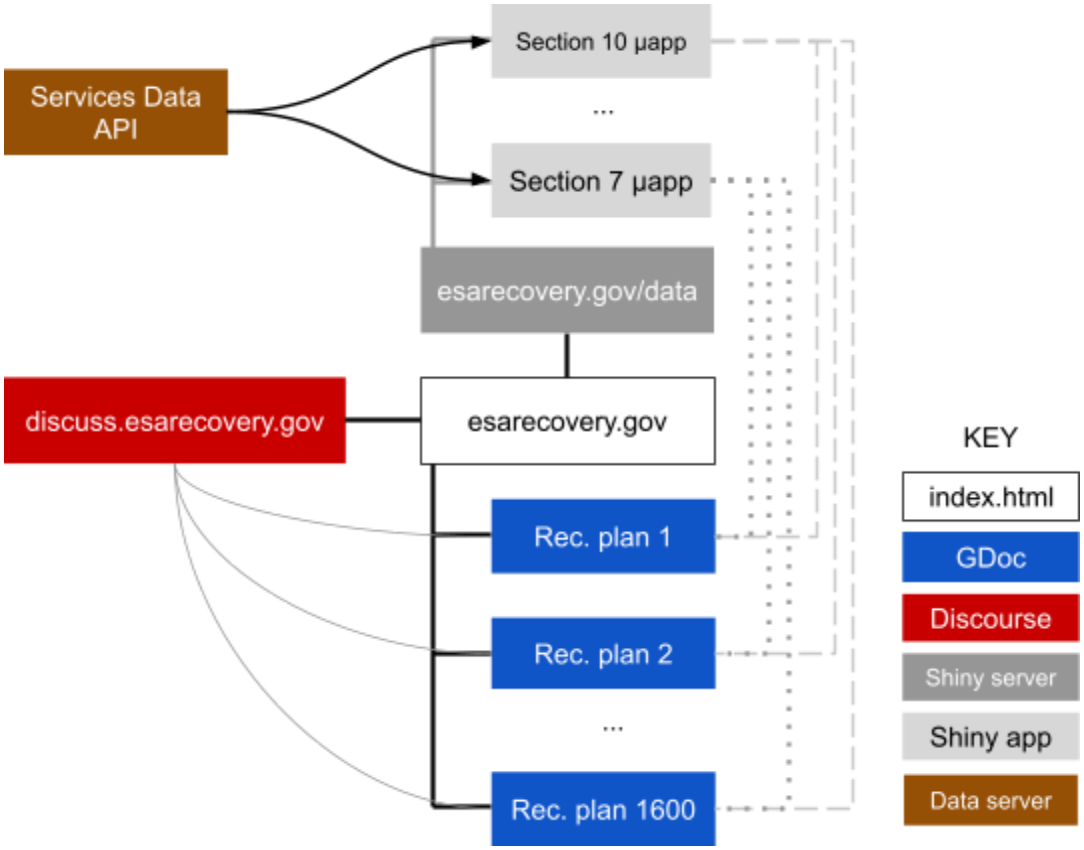
WordPress

The appearance of the WordPress-based dynamic recovery plan is most similar to the fully customized prototype developed in Shiny, but holds many advantages. Because it is so widely used, WordPress is extremely well-developed and there is a huge community that contributes to its continued development. Security is all-but solved, and - importantly for the Services - can use authentication with GSuite applications. The standard page and post editor is a fully developed rich text editor, with familiar buttons, so Services personnel will readily recognize. Plugins can be easily used to embed dynamic graphs, maps, and tables.

Discussion

Recovery plans are essential for guiding ESA-listed species back from the brink of extinction. But current recovery plans that are based on static documents - typically written and approved once, then rarely updated - are not taking advantage of new tools that can improve recovery planning and implementation. Our goal was to evaluate possible solutions to the shortcomings of static recovery plans, that is, to identify one or more

"ideal" platforms for dynamic recovery plans. Despite the option being right in front of our faces, we didn't realize that we would find GDocs is the best solution because we weren't looking there. The Services already use GSuite in their operations, Google has already deployed GSuite at global scale, and the platform appears to be plenty sufficient and offers the best combination of features for minimal cost. We recommend that the Services use GDocs for the core of modern, web-based, and dynamic recovery plans. We further recommend that the best features of the Shiny and Discourse platforms be used to enhance the GDoc-based recovery plans[2]. The recovery platform might be summarized as:



The option of using GDocs for dynamic recovery plans only came up a couple of days before completing this evaluation. After setting up a custom git server to illustrate how a self-hosted GitBook solution could work, we began thinking through the details of how people - in particular personnel from the Services - would use these plans in their day-to-day work. And it was then that we realized the extent to which our technical proclivities were biasing our view of how dynamic recovery plans could or should be implemented. The software development crew at Defenders uses git and markdown every day, so these tools that are part of the Shiny, GitBook, and/or Discourse are completely familiar...but likely alien to most other people. When starting to write down, *in GDocs*, this "revelation" about the gap between developer and user skills, and the steep learning curve

[2] R programming with Shiny "glue" to the web is the ideal way to handle and present data. While commenting directly on GDocs is possible, Discourse is perfectly suited to actual conversations.

for parts of the first three plan types, we realized that GDocs might be the best possible choice. At that point we set up the scoring criteria and revisited each plan type to critically evaluate each platform, which led to the conclusions we have presented above.

If the Services were to take this recommendation and use WordPress to modernize recovery plans, what are the implications for Defenders' involvement in the project? Most of our time to this point has been spent on software development, and we anticipated needing many more months of development. (And a significant amount of time has been focused on what kinds of new information can be directly integrated in the plans.) But with one minor exception[3], there would be no need for us to further develop the system: WordPress engineers and the WordPress community, who have vastly more resources than the Services or Defenders, will continue doing that. We suggest that rather than developing a novel recovery plan software system, Defenders can focus on helping the Services convert existing recovery plans to dynamic recovery plans as needed. This would entail several components.

First, we would work with the Services to make the "data core" of dynamic recovery plans publicly available through a unified API. The data that needs to be included are:

- the relevant fields (section 7) of TAILS and PCTS databases;
- the Recovery Online Activity Reporting (ROAR) database, which is jointly administered by the Services;
- reported ESA expenditures data; and
- data on section 10 conservation agreements.

We have acquired snapshots of the first two pieces of data from the Services by requesting data dumps, and the third by extracting data from annual expenditure reports. We acquired the fourth data type by scraping all section 10 agreement tables from FWS's ECOS website, where we found significant inconsistencies between portals (see whitepaper). Each of these pieces of data is already collected by the Services and, per President Obama's Executive Order on publicly available and machine readable data, should be publicly available and machine readable. The transition to dynamic recovery plans that integrate real-time data with fundamental knowledge about listed species and the actions needed for their recovery is an ideal motivator.

Second, we would work with the Services and other interested parties - in particular Department of Defense, who has expressed high interest in dynamic recovery plans - to develop the initial list of species whose dynamic recovery plans we will convert or create. We think this should include at least one species in each of the following categories:

- complex recovery scenarios (e.g., multi-jurisdiction);

[3] We should create a simple tool to scrape current tables/graphs/maps for the core data components of dynamic recovery plans (e.g., distribution data, section 7 consultations, section 10 agreements). This task is minor compared to the much larger task of creating a novel system.

- severely out-of-date plans (e.g., indigo snake, 1982 plan);
- an ecosystem plan (e.g., Yaqui fishes);
- a recovery outline (e.g., NMFS's outline for five Caribbean corals);
- needing recovery actions primarily from non-federal entities (?example?); and
- one of the >400 species currently lacking a plan.

We anticipate that completing a dynamic recovery plan in each of these categories will offer new learning opportunities that can be applied to most of the nearly 1,600 plans that are needed.

Third, we would work with the Services to develop the infrastructure needed to collect, display, and integrate with dynamic recovery plans currently "missing data." This includes data on:

- amount of take authorized through section 7 consultations;
- changes of each species' threat and biological status (see Malcom et al. 2016); and
- the relative importance (past and future) of the threats each species faces.

We recognize that other parties may find additional data types that would be particularly useful when using, commenting on, and contributing to the dynamic recovery plans. We are open to evaluating how much Defenders would invest in developing these alternative data

Fourth, we would work closely with the Services and with the communities using the dynamic recovery plans to critically evaluate the effectiveness, strengths, and weaknesses of the plans several (3-5) years in the future. This review might follow the structure of NMFS's recent Recovery Program Review; the details of evaluation metrics would be developed closer to the time of such a review.

Conclusion

The Services recognize the need to modernize recovery planning. Our work suggests that they can leap ahead with dynamic recovery plans by using existing, nearly free tools that add very little cost to the recovery program but offer huge benefits. We are happy to discuss the details - including options not discussed above - at any time.