Free Tools—Not So “Free”? The Evolution, Risks and Costs of Fishery Stock Assessment Tools

Natalie Anne Dowling[1], Catherine Mary Dichmont [2,3], André Eric Punt[1,4], Sean Pascoe[5], Roy Aijun Deng[5], Jason Marc Cope[6], Elizabeth F. Perl[7]

SEDAR103-RD-32

Fish and Fisheries

WILEY

FISH and FISHERIES

ORIGINAL ARTICLE OPEN ACCESS

Free Tools—Not So "Free"? The Evolution, Risks and Costs of Fishery Stock Assessment Tools

Natalie Anne Dowling[1] | Catherine Mary Dichmont[2,3] | André Eric Punt[1,4] | Sean Pascoe[5] | Roy Aijun Deng[5] | Jason Marc Cope[6] | Elizabeth F. Perl[7]

[1]Environment, CSIRO, Castray Esplanade, Hobart, Tasmania, Australia | [2]Cathy Dichmont Consulting, Banksia Beach, Queensland, Australia | [3]Institute for Marine and Antarctic Studies, University of Tasmania, Battery Point, Tasmania, Australia | [4]School of Aquatic and Fishery Sciences, University of Washington, Seattle, Washington, USA | [5]Environment, CSIRO, Queensland Bioscience Precinct, St Lucia, Queensland, Australia | [6]NOAA Fisheries, Northwest Fisheries Science Center, Seattle, Washington, USA | [7]ECS Federal in Support of NOAA Fisheries Office of Science and Technology, Silver Spring, Maryland, USA

Correspondence: Natalie Anne Dowling (natalie.dowling@csiro.au)

Received: 25 September 2025 | **Revised:** 6 February 2026 | **Accepted:** 23 February 2026

Keywords: fisheries | software tools | stock assessment | tool development | tool evolution

ABSTRACT

In fisheries stock assessment, free software tools have proliferated in recent years. These tools provide standardised and, in many cases, well-tested and efficient ways to conduct and support stock assessments, making cutting edge expertise globally accessible. However, they come with considerable ongoing costs, beyond their conceptual development and coding. We seek to understand and analyse the process by which fisheries stock assessment tools have evolved and identify the risks and challenges associated with their development, general use, and ongoing maintenance. We postulate on the development process and risk points for the endurance of fisheries science software tools, using theory related to volunteering. To confront our expectations with empirical evidence, we undertook a survey of stock assessment tool developers. Overwhelmingly, despite moderate institutional support, developers work beyond regular hours, especially when upscaling a tool to wider use, providing updates to the tool, and conducting maintenance activities. Notably, our expectation that tools with limited employer/institutional support do not survive was only partially supported—we found many tools persist despite limited support, primarily through substantial voluntary effort from developers. Future efforts for tool development should focus on (i) better integrating tool development and maintenance into institutional frameworks; (ii) better appreciating the motivation of and need for free tools; (iii) as such, advocating for their development and, particularly, ongoing maintenance; and (iv) for these activities to become internalised and owned in an ongoing manner as part of core business.

1 | Introduction

Scientific software tools (Kelly 2015) that are freely available online have become common in many fields, including fisheries science. We define "software tools" to include code to run algorithms, estimate parameters by fitting models to data, produce diagnostics, build guidance websites, and apply graphical user interfaces. In fisheries stock assessment, such tools may form the basis for stock assessments or assist their accessibility, application, and interpretation. Many such tools are "free" in that users pay nothing to access and utilize them. We take this "price" definition hereafter, as opposed to the "liberty" definition of "free software" (https://www.gnu.org/philosophy/free-sw.html), whereby users have freedom to run, copy, distribute, study, change and improve the software—though many of these freedoms may also apply.

https://doi.org/10.1111/faf.70076

Free software tools have made cutting-edge expertise globally accessible, including providing standardised, generally well-tested and efficient ways to conduct stock assessments and provide management advice, particularly when capacity is lacking. Within fisheries science, they provide communication conduits to global experts, and link fisheries and organisations across common software, user groups and training platforms. While tools that are so user-friendly that they can be applied without critical thought may reduce innovation through rote application, they also can achieve the opposite if they are open source, as people use and continually contribute to them. Free software has reduced the need for redundant re-coding and re-testing of methods (although duplication does still occur, particularly when tools are not well known), saving time and reducing the risk of coding errors or making the mistakes others have already made and learnt from. The most successful of these tools are globally recognised and represent cutting-edge methodologies/approaches, with many incorporating other open science aspects (e.g., user groups; version control and the use of collaboration software tools such as GitHub (https://github.com/); ongoing training).

Free tools come with considerable ongoing costs (Dichmont et al. 2016), beyond their inception and initial coding. Online tools continue to evolve, unlike methods only described in published papers. Demand for extensions increases as tools become used by broader communities. However, tools can also stagnate, become incompatible with newer software versions, rely on unsupported dependencies, or become obsolete without continued maintenance.

To date, little attention has been paid in fisheries science to understanding the process by which stock assessment tools have evolved, and the risks and challenges associated with their development, use, and ongoing maintenance. To address this gap, we outline our expected development process and risk points for these tools, drawing on theory related to small business development and expansion (Churchill and Lewis 1983; Scott and Bruce 1987) and volunteering (Hustinx et al. 2010).

2 | Theoretical Framework and Study Aims

Understanding tool evolution requires consideration of how software is typically developed. The lifecycle of software development generally involves (a) describing requirements, (b) planning, (c) designing, (d) implementing prototypes and development, (e) testing, (f) delivery and (g) maintenance (Hossain 2023; Kruchten 2004; Boehm 1988; Royce 1970). These stages can occur iteratively (by starting small and building additional components), sequentially, or cyclically (in and out of stages) (Schwaber and Beedle 2002). We suggest that this lifecycle can apply to stock assessment software tools, but that few developers and employers (agencies/institutions/organisations) recognise this and undertake formal development and maintenance processes. We suggest tools rather follow a lifecycle of origin (approximately analogous to points (a–c) above), transition (points (d–e)), upscaling (per point (f)), maintenance (point (g)) and succession phases, each with associated costs.

Stock assessment tool development differs from commercial software development in that the tools are meant to be accessible and usually free to a broader community. Developing tools demands time, talent, and resources without expectation of recouping costs or making money. An important consideration is the source of the original and ongoing funding and support. We seek to empirically understand whether development and maintenance are undertaken at the developer's behest and the extent to which developer time and effort can be viewed as volunteerism or a “donation” to the field. Volunteerism is diversely defined (Hustinx et al. 2010). Many individuals consider work-related activities out of official work hours as volunteerism, while others consider it an expected work activity (Handy et al. 2000). When developing stock assessment tools, some overtime may be considered normal and important to science, but excessive periods are not and are potentially unsustainable.

Volunteerism extends beyond developers with the wide adoption of open-source software practices. Stock assessment tools increasingly use Git and GitHub to provide version control, distribute releases and documentation, engage with users, and allow user contributions. Tool developers typically value user assistance with development and maintenance tasks. Developers can onboard inexperienced volunteer contributors who ultimately become experienced and valued contributors. By inviting volunteer contributions, developers gain new ideas, faster development and other benefits from open-source software development. However, relying on users as volunteers puts tools at risk if those volunteers become unable to continue contributing (Schweik and English 2007) and introduces opportunity for errors if volunteers are not adequately supervised.

We expect that key risk points to tool survival occur where developers have larger expectations for their tools than their employer or funding model, or vice versa. We further expect that developers lag on tool maintenance when their tool lacks ongoing funding, personnel/capacity, or a developer community. An associated risk point may be a lack of formal programming expertise: stock assessment scientists do not necessarily code as effectively as formally trained programmers or software developers (Thimbleby 2024). Simply, we expect that tools lacking support—specifically, ongoing institutional support in terms of funding and time that is formally acknowledged—do not survive, and those that are supported do.

In this study, we survey tool developers to obtain empirical evidence confronting our expectations about tool lifecycles, risk points, and the role of volunteerism in tool development and maintenance. We seek to gauge the extent of support for the development, upscaling and maintenance of freely available software tools, and identify whether and how any of these steps could become easier and more sustainable. We draw parallels with volunteerism (Dutta-Bergman 2004) to confront risks and challenges. Given substantial costs to developers, we provide advice on supporting and managing tool development, upscaling, and ongoing maintenance. Ultimately, we hope to build appreciation of what it takes to build and maintain software tools within fisheries science, and advocate for their proper support. We provide recommendations to ensure consequent risks are reduced and freely available online stock assessment tools are more explicitly acknowledged as important contributions.

3 | Methods

3.1 | Confronting Expectations on Tool Development With Empirical Data

We administered a survey to tool developers (Data S1), asking about the history and evolution of their free fisheries science tools, and sought to identify the risk points. We considered the following issues:

1. Inception: the impetus for the tool, initial development funding and duration
2. Transition (as applicable) (the point at which a tool moves from a concept and initial coding of a pilot or local version to a "generalisable" standalone tool that can be made available to, and utilised, by others): (i) process and resourcing for the transition to a generalisable tool; and (ii) whether developers actively upscale tools from personal projects to generalised software, and if so, the drivers and duration of this phase
3. Upscaling (as applicable): (i) how the tool expanded; and (ii) when and how it became well known (promoted) and used by others
4. Maintenance: (i) process and funding; and (ii) why a tool did not endure (if applicable)
5. Succession: how a tool endures once the main developer is no longer involved
6. Obsolescence

3.2 | Selection of Tools and Developers

We considered 34 freely available online software tools (reduced to 33 on the basis of received survey responses) (Table 1), based on the list of international fishery stock assessment software packages collated in the Australian Stock Assessment Toolbox (Dichmont et al. 2021), and expanded to include key stock assessment diagnostic tools, guidance websites, decision support tools, broader ecosystem models, and graphical user interface tools. While all reasonable efforts were made, we do not claim that our final list of tools was exhaustive, particularly across the expanded inclusion categories.

Project team members made direct email approaches to the primary developer of each tool, or to a co-developer, depending on individual availability. In some cases, developers requested that additional people, either associated with their tool or who had developed other tools, be invited to participate in the survey, and we subsequently did so.

3.3 | "Free Tools" Considered

The freely available online tools we considered here (Table 1) fell under the following broad themes:

- Stock assessment packages, ranging from data-limited methods to fully integrated models that include ecosystem effects. Most are designed for single-species stock assessments, though some address multispecies systems. Packages may include a single or a suite of methods.
- Adjunct tools that add features or diagnostics to support integrated assessment packages
- Tools that enable management strategy evaluation (MSE) of alternate management procedures or harvest strategies
- Ecosystem modelling packages
- Decision support tools that guide harvest strategy development and selection among stock assessment approaches

A brief overview of each tool is provided in Data S2, highlighting key features and providing broad guidance for potential applications. Regardless of their region of origin, all packages have global relevance.

For the stock assessment packages, more detailed guidance on how to navigate and select among them is provided by two (included) decision support tools—*FishPath* and the *Stock Assessment Toolbox*.

3.4 | Survey Design

The survey comprised 51 multiple choice and open ended questions (Data S1), organised into seven themes, intended to capture our theorised tool evolutionary process and life cycle: (a) technical/general; (b) the origin of the method/idea/concept; (c) how the transition to a generalisable tool was undertaken and resourced; (d) upscaling to a broader user group; (e) maintenance; (f) succession; and (g) obsolescence. 35 questions were multiple choice (with some allowing multiple responses), but with no option for participants to add extra explanatory text, nor a "not applicable" option. Some questions were conditional on a certain response to a previous question (e.g., if respondents indicated that they used version control, they were then asked when they did so). Seven multiple choice questions included a response option of "other", and provision for the respondent to elaborate. Nine questions required prose responses.

The survey was administered using the Qualtrics online survey platform (https://www.qualtrics.com/). A link was sent with the email request for participation; respondents submitted their responses online.

Ethics approval for the survey was provided by the CSIRO Social Science and Human Research Ethics Committee (approval 001/23).

4 | Results

4.1 | Survey Responses

We received responses from each developer that we approached, and in total obtained 40 survey responses (we received additional requests beyond the original 34 individuals approached). Five surveys were incomplete (25%–62% questions answered),

TABLE 1 | List of stock assessment tools considered in this paper and for which survey results were obtained, together with the key reference for each, the current link to the software, the region in which the tool was developed (noting that all have global applicability), the species for which the tool was developed (where relevant), the system accommodated by the tool, and the type of tool.

Tool name	References	Link to software	Full name	Region in which tool was developed NOTE: all have global application	Species (where relevant)	System	Type
Barefoot Ecologist	Prince (2003)	http://barefootecologist.com.au/	The Barefoot Ecologist	Australia		Single species	Length-based assessment
LIME	Rudd and Thorson (2017)	https://github.com/merrillrudd/LIME	Length-based Integrated Mixed Effects	USA		Single species	Length-based assessment
datalowSA	Haddon et al. (2019)	https://github.com/haddonm/datalowSA	Data Low Stock Assessment	New Zealand		Single species	Suite of data-limited assessments
MQMF	Haddon (2021)	https://github.com/haddonm/MQMF	Modelling and Quantitative Methods in Fisheries	Australia		Single species	Suite of data-limited assessments
TropFishR	Mildenberger et al. (2017)	https://github.com/tokami/TropFishR	Tropical Fisheries Analysis	Europe: Food and Agriculture Organisation of the United Nations (FAO)		Single species	Suite of data-limited assessments
JABBA	Winker et al. (2018)	https://github.com/jabbamodel/JABBA	Just Another Bayesian Biomass Assessment	South Africa and the Pacific Islands Fisheries Science Center (NOAA, USA)		Single species	State space surplus production model
SPiCT	Pedersen and Berg 2017	https://github.com/DTUAqua/spict	Stochatic Surplus Production model in Continuous Time	Denmark		Single species	Surplus production model
a4a	Jardim et al. (2014)	https://github.com/flr/FLa4a	Assessment for All	Developed for the European Union		Single species	Simple statistical catch-at-age model
MULTIFAN-CL	Fournier et al. (1998)	https://mfcl.spc.int/	Multiple Frequencies of Analysis—Catch at Length	South Pacific		Single species	Fully integrated assessment
SAM	Nielsen et al. (2021)	https://github.com/fishfollower/SAM	State-space Assessment Model	European Union		Single species	Fully integrated assessment

(Continues)

TABLE 1 | (Continued)

Tool name	References	Link to software	Full name	Region in which tool was developed NOTE: all have global application	Species (where relevant)	System	Type
WHAM	Stock and Miller (2021)	https://github.com/timjmiller/wham	Woods Hole Assessment Model	USA		Single species with environmental effects	Fully integrated assessment
CASAL2	Casal2 Development Team (2023)	https://github.com/Casal2	C++ Algorithmic Stock Assessment Library	New Zealand		Single/multi species	Statistical age- or size-structured fully integrated assessment
AMAK	Ianelli (2022)	https://github.com/afsc-assessments/AMAK	Assessment Model for Alaska	Alaska, USA		Single species	Statistical catch-at-age fully integrated assessment
Stock Synthesis	Methot Jr and Wetzel 2013	https://nmfs-ost.github.io/ss3-website/	Stock Synthesis	USA		Single species with environmental effects	Statistical catch-at-age fully integrated assessment
SAC (formerly SS-DL)	Cope (2020); Cope (2024)	https://github.com/shcaba/SS-DL-tool	Stock Assessment Continuum	USA		Single species	Continuum of (data-limited through to data-rich) assessment approaches
SS3 Model Diagnostics	Carvalho et al. (2021)	https://github.com/jabbamodel/ss3diags	Stock Synthesis 3 Diagnostics	USA		Single species	diagnostics of stock assessment output
SS3Sim	Anderson et al. (2014)	https://github.com/ss3sim/ss3sim	Stock Synthesis Simulation Output	USA		Single species	Simplication of simulation output generation
R4ss	Taylor et al. (2021)	https://github.com/r4ss/r4ss	R code for Stock Synthesis	USA		Single species	Support tool to work with Stock Synthesis input and output files
GMACS	unpublished	https://github.com/GMACS-project/GMACS_tpl-cpp_code	General Model for Assessing Crustacean Stocks	Alaska, USA	Focused on crustaceans (originally designed for Alaska crab stocks)	Single species	Size structured integrated assessment

(Continues)

TABLE 1 | (Continued)

Tool name	References	Link to software	Full name	Region in which tool was developed NOTE: all have global application	Species (where relevant)	System	Type
GMR	Veron and Szuwalski (2024)	https://gmacs-project.github.io/gmr/	A Generalised size-structured Model for Assessing Crustaceans in R	USA	Focused on crustaceans	Single species	Size structured integrated assessment
VAST	Thorson (2019)	https://github.com/James-Thorson-NOAA/VAST	Vector Autoregressive Spatio-Temporal (VAST) package	USA		Multispecies	Spatial delta-generalised linear mixed model (delta-GLMM)
sdmTMB	Anderson et al. (2024)	https://pbs-assess.github.io/sdmTMB/	Species Distribution Models with TMB	USA		Single species	Generalised linear mixed effects models
CEATTLE	Holsman et al. (2016) Adams et al. (2022)	https://github.com/grantdadams/Rceattle	Climate-Enhanced, Age-based model with Temperature-specific Trophic Linkages and Energetics	Bering Sea, USA	Groundfish	Multispecies	Multispecies age-structured assessment model
GADGET	Stefansson and Palsson (1998)	https://www.hafogvatn.is/static/research/files/fjolrit-058.pdf https://www.hafogvatn.is/static/research/files/fjolrit-059.pdf https://www.hafogvatn.is/static/research/files/fjolrit-060.pdf	Globally Applicable Area Disaggregated General Ecosystem Toolbox	Iceland: designed for ICES fishers	Cod and capelin as an example	Multispecies with ecosystem	Fully integrated assessment/ ecosystem model
GADGET2	Begley (2006) Þór Elvarsson et al. 2018	https://github.com/gadget-framework/gadget2	Globally Applicable Area Disaggregated General Ecosystem Toolbox	European Union: designed for ICES fisheries		Multispecies with ecosystem	Fully integrated assessment/ ecosystem model

(Continues)

TABLE 1 | (Continued)

Tool name	References	Link to software	Full name	Region in which tool was developed NOTE: all have global application	Species (where relevant)	System	Type
GADGET3	Lentin et al. (2025)	https://github.com/gadget-framework/gadget3/	Globally Applicable Area Disaggregated General Ecosystem Toolbox	European Union: designed for ICES fisheries		Multispecies with ecosystem	Fully integrated assessment/ecosystem model
FLR	Kell et al. (2007)	https://github.com/flr/FLAssess	Fisheries Libraries for R	European Union: designed for ICES fisheries		Single species	Management strategy evaluation (MSE)
OpenMSE	Hordyk et al. (2022)	https://github.com/Blue-Matter/openMSE	Open Source Software for Management Strategy Evaluation	Canada		Single species (though can use the function multiMSE() to conduct MSE for multiple fleets/stocks)	Management strategy evaluation (MSE)
SSMSE	Doering et al. (2023)	https://github.com/nmfs-fish-tools/SSMSE https://nmfs-fish-tools.github.io/SSMSE/	Stock Synthesis Management Strategy Evaluation	USA		Single species	Management strategy evaluation (MSE)
ATLANTIS	Fulton et al. (2011)	https://github.com/Atlantis-Ecosystem-Model		Australia		Ecosystem	Ecosystem model
Rpath	Lucey et al. (2020)	https://noaa-edab.github.io/Rpath/articles/Rpath.html	Rpath: an open source food web model	USA		Ecosystem	Ecosystem model with management strategy evaluation (MSE)
FishPath	Dowling et al. (2023)	https://fishpath.org/		Australia & USA		Single and multiple species	Decision support tool for harvest strategy development
Stock Assessment Toolbox	Dichmont et al. (2021)	https://toolbox.frdc.com.au/	Stock Assessment Toolbox	Australia; emphasis on Australian application		Single and multiple species	Decision support tool for stock assessment package selection

and two respondents did not provide consent. These seven responses were excluded, yielding a final sample size of 33. Respondents who did not complete the survey showed no obvious distinguishing characteristics based on submitted information. The small sample size ($n = 33$) limits the generalisability of specific responses, although broader patterns across the lifecycle stages are more robust.

The full survey and details of responses are provided in Data S1. Figure 1 provides a conceptual overview of tool development across the full lifecycle. We conducted comprehensive statistical tests to examine relationships between tool characteristics (type, funding status, employer support), lifecycle stages, and key outcomes (maintenance hours, outside hours work, succession planning, obsolescence). All statistical tests are summarised in Table 2. Detailed findings for each lifecycle stage are presented below.

4.2 | Summary of Responses

4.2.1 | Tool and Developer Characteristics

Most tools were stock assessment packages (24), MSE tools (14), or diagnostic tools (11), with the remainder decision support tools (10) or other (8) (noting tools could straddle types). Tool type did not significantly predict hours spent on maintenance (Kruskal–Wallis test, $p = 0.93$), self-funding rates (Fisher's exact test, $p = 0.69$), obsolescence risk ($p = 0.67$), outside hours work ($p = 0.16$), or succession planning rates ($p = 0.35$), though small sample sizes for non-stock-assessment tools limit statistical power (Table 2).

Most tools were written in R/RCpp (15), TMB (4), ADMB (2), or combinations thereof (7), with 91% using formal version control, many from early during development. 45% of respondents were sole leads without support teams, while 39% led with team support. The remaining five respondents were part of the support team for their tool, but were not the main developer.

Critically, 40% of respondents indicated that tool development, updating, and maintenance exceeded expectations for their regular role, while 28% characterised their tool as a “passion project” undertaken beyond work obligations (Figure 2a). Another 28% indicated this work was part of regular in-hours project work. Regardless, 88% of respondents felt their tool was used at least moderately, with 30% reporting extensive use.

4.2.2 | Inception

The original motivation for tools was most commonly a current project or program need (39%), while 27% originated as passion projects (Figure 2b). At inception, employer interest/involvement was split between none/low (30%), moderate (39%), and high (30%) (Table S1). 45% of tools had full funding at conception, while 30% had no funding (Figure 1). Tools that began unfunded were more likely to be viewed by developers as personal passion projects (60%) compared to tools with initial funding, though not significantly so (30%; Fisher's exact test, $p = 0.14$) (Figure 1 and Table 2). However, inception funding did not predict self-funding rates at maintenance (40% vs. 44%, Fisher's exact test, $p = 1.0$), obsolescence (30% vs. 30%, $p = 1.0$), or outside hours work during transition or maintenance (Mann–Whitney *U* tests, all $p > 0.11$) (Table 2).

4.2.3 | How the Transition to a Generalisable Tool Was Undertaken and Resourced

Time between recognising potential value and developing tools for broader use varied from at inception to over 10 years. Transition was funded by respondents' employers in over half (58%) of cases, with six tools self-funded as passion projects. Two respondents noted delayed funding contracts or retrospective funding following proof-of-concept demonstration.

At generalisation, employer interest was moderate for 48% of respondents, low for 25%, and high for 28%. Almost half (48%) of respondents felt their employer granted flexibility to develop their tool as core business, though 12% felt they had no such flexibility.

Despite this, 60% of respondents indicated at least “a moderate amount” of transition work occurred outside regular hours (Figure 1), with 15% stating “most” occurred out-of-hours (Figure 3a). Only 15% stated none of the transition work occurred out-of-hours. There was a moderate negative correlation between employer support at transition and outside hours work (Spearman's rho $= -0.30$, $p = 0.094$) (Figure 1 and Table 2).

Transition for 55% of respondents occurred as part of project work, while for 27% it continued to be or became a passion project. 24% developed generalised tools alone, 33% within institutional teams, and 42% in cross-institutional teams. 30% of respondents needed software developers, 9% needed application/interface expertise, and 15% sought additional conceptual expertise.

4.2.4 | Upscaling: When It Became Well Known and/or Used by Others

Time to widespread adoption ranged from within a year to five or more years, with 24% reporting gradual uptake over 2–3 years. Workshops/working groups (18%), conferences (12%), and teaching opportunities (9%) proved effective for dissemination, alongside online publication (12%), literature publication (21%), and successful case studies (21%).

The majority (79%) of respondents remain actively involved in applications of their tool. 57% reported their tool has been cited in over 10 publications, while only three reported no citations. 39% indicated other packages depend on their tool or were developed based on it, with an additional 30% uncertain whether this occurred. 58% of respondents have had their tool formally peer reviewed. 55% can report usage metrics, though many rely on platforms like GitHub rather than actively tracking metrics themselves.

Upscaling was generally funded by respondents' employers (47%), or self-funded (34%). For 12% of respondents, their employer

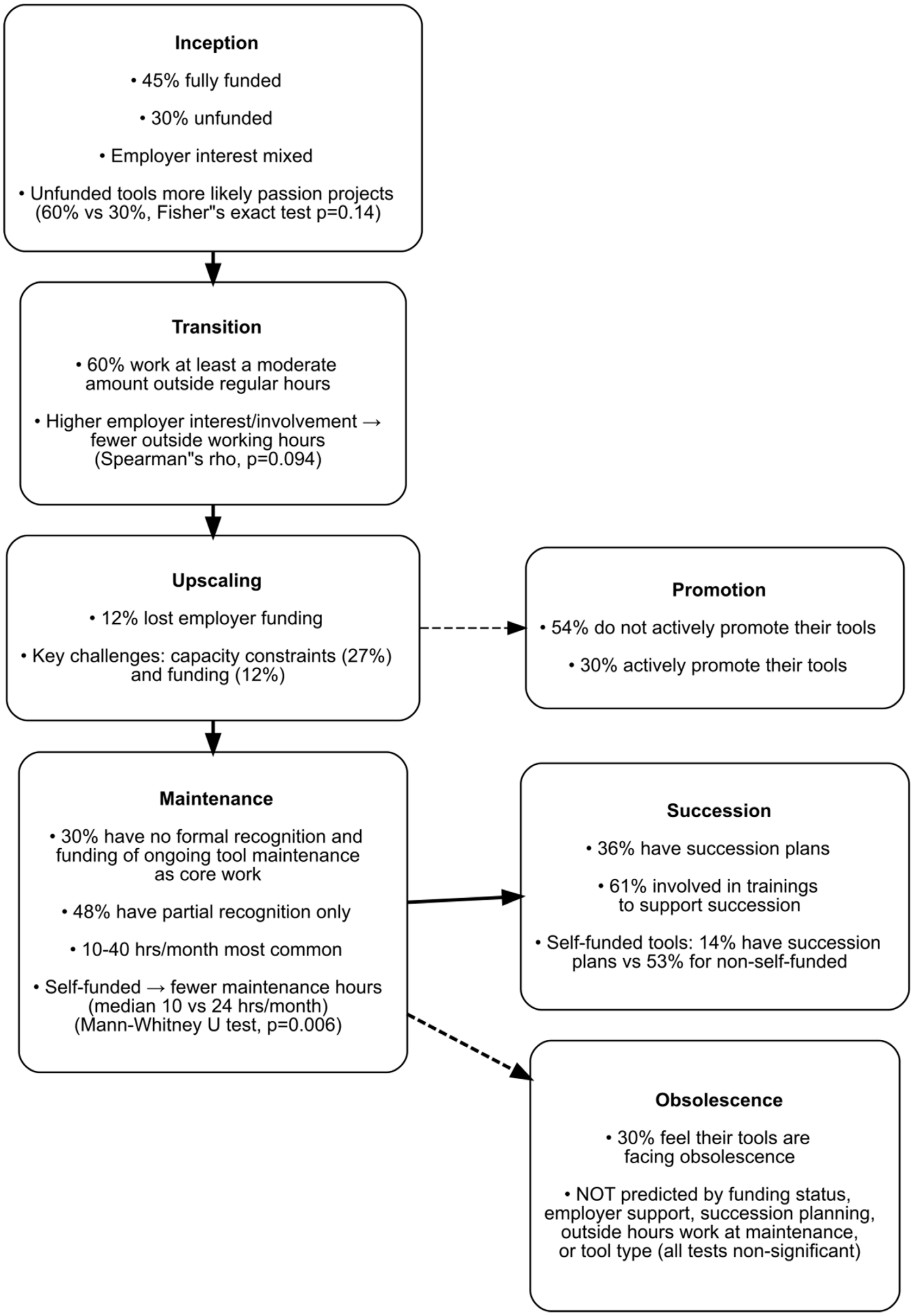

FIGURE 1 | Conceptual lifecycle of fisheries software tool development, with some key findings at each stage.

no longer funded the tool at upscaling, so it became self-funded (Figure 1). Employer interest at upscaling remained mixed: 34% perceived low interest, 44% moderate, and 22% high. However, 63% of respondents reported their employers accepted and supported their involvement. 31% of respondents felt some lack of prioritisation or level of resistance from their employer; all of these had also reported low or moderate institutional interest/involvement.

Most respondents (64%) did not need additional expertise beyond themselves or their team at upscaling, although 12% brought in software developers and 21% brought in application/interface expertise. Only 6% experienced no challenges upscaling. 27% cited capacity constraints and 12% cited funding as main challenges (Figure 1).

Consistent with these challenges, 54% do not actively promote their tools (Figure 1), citing limited capacity, lack of need, or shifting focus. Active promotion (30% of respondents) (note, there were additional unclear or "N/A" responses) occurred mainly through in-person engagements (teaching, workshops, office hours, presentations). Almost all (94%) have trained others to use their tool.

TABLE 2 | Statistical tests examining tool lifecycle patterns. Tests are grouped by outcome variable (what is being predicted or compared).

Hypothesis/question tested	Variables/comparison	Test	Sample size	Result	*p*
Funding patterns across lifecycle					
Self-funding increases from transition to maintenance	Self-funding rates: Transition vs. Maintenance	McNemar's test	$n=32$	Self-funding increased from 18% to 42%	0.013*
Tools shift from institutional to self-funding over lifecycle	Individual tool funding trajectories from inception to maintenance	Descriptive	$n=32$	31% shifted institutional→self-funded; 19% shifted self→institutional	—
Predictors of self-funding at maintenance					
Unfunded tools at inception become self-funded at maintenance	Inception funding status vs. self-funding at maintenance	Fisher's exact	$n=33$	No significant association (40% vs. 44% self-funded)	1.0
Lack of formal recognition leads to self-funding	Employer recognition level vs. self-funding at maintenance	Fisher's exact	$n=33$	No significant association (60% vs. 35%–33%)	0.53
Tool type affects self-funding rates	Tool type vs. self-funding status	Fisher's exact	$n=33$	No significant association	0.69
Employer recognition patterns					
Employer recognition/support remains stable across lifecycle	Mean support/recognition levels across 4 lifecycle stages	Friedman test	$n=33$	Stable across stages (mean 1.88–2.03)	0.61
Predictors of maintenance hours invested					
Self-funding reduces maintenance hours invested	Self-funded vs. non-self-funded tools: maintenance hours	Mann–Whitney *U*	$n=25$	Self-funded: fewer hours (median 10 vs. 24)	0.006*
Institutional recognition affects maintenance hours invested	Recognition level vs. maintenance hours per month	Kruskal–Wallis	$n=25$	No significant difference (median 10–25 h)	0.71
Tool type affects maintenance hours required	Tool type vs. maintenance hours	Kruskal–Wallis	$n=25$	No significant difference	0.93
Predictors of outside hours work					
Outside hours work persists from transition to maintenance	Outside hours work: Transition vs. Maintenance	Wilcoxon signed-rank	$n=32$	No significant change (60% vs. 55% moderate or more)	0.61
Higher employer support reduces outside hours work at transition	Support level vs. outside hours work at transition	Spearman's rho	$n=33$	Moderate negative correlation (rho = −0.30)	0.094
High employer support reduces outside hours work at transition	High vs. low employer support at transition	Ordinal regression	$n=33$	Lower outside hours (coefficient = −1.58)	0.095
Higher employer support reduces outside hours work at maintenance	Support level vs. outside hours work at maintenance	Spearman's rho	$n=31$	Moderate negative correlation (rho = −0.28)	0.12

(Continues)

TABLE 2 | (Continued)

Hypothesis/question tested	Variables/comparison	Test	Sample size	Result	p
Full employer recognition reduces outside hours work at maintenance	Full vs. no recognition at maintenance	Ordinal regression	$n=31$	Lower outside hours (coefficient = −1.67)	0.090
Lack of formal recognition increases outside hours work	Recognition level vs. outside hours work at maintenance	Spearman's rho	$n=31$	Weak negative correlation (rho = −0.23)	0.20
Inception funding affects outside hours work during transition/maintenance	Funded vs. unfunded at inception vs. outside hours	Mann–Whitney U	$n=33$	No significant difference	>0.11
Tool type affects outside hours work	Tool type vs. outside hours work	Kruskal–Wallis	$n=33$	No significant difference	0.16
Predictors of succession planning					
Self-funding reduces succession planning	Self-funded vs. non-self-funded tools: succession plans	Fisher's exact	$n=33$	Self-funded: lower succession (14% vs. 53%)	0.033*
Employer support increases succession planning	Support level vs. presence of succession plan	Kruskal-Wallis	$n=33$	No significant difference across lifecycle stages	>0.20
Tool type affects succession planning	Tool type vs. succession planning	Fisher's exact	$n=33$	No significant association	0.35
Predictors of obsolescence					
Self-funding increases obsolescence risk	Self-funded vs. non-self-funded tools: obsolescence	Fisher's exact	$n=33$	No significant association	0.71
Succession planning reduces obsolescence risk	Succession plan vs. obsolescence	Fisher's exact	$n=33$	No significant association	1.0
Unfunded tools at inception face obsolescence	Inception funding status vs. obsolescence	Fisher's exact	$n=33$	No significant association (30% vs. 30% obsolescent)	1.0
Outside hours work prevents obsolescence	Outside hours work vs. obsolescence	Mann–Whitney U	$n=31$	No significant association	0.43
Employer support prevents obsolescence	Support level vs. obsolescence across stages	Kruskal-Wallis	$n=33$	No significant difference	>0.23
Tool type affects obsolescence risk	Tool type vs. obsolescence	Fisher's exact	$n=33$	No significant association	0.67
Passion projects					
Unfunded tools at inception are passion projects	Inception funding status vs. passion project classification	Fisher's exact	$n=33$	Trend: unfunded more likely (60% vs. 30%)	0.14

Note: Asterisks (*) indicate statistically significant results ($p<0.05$).

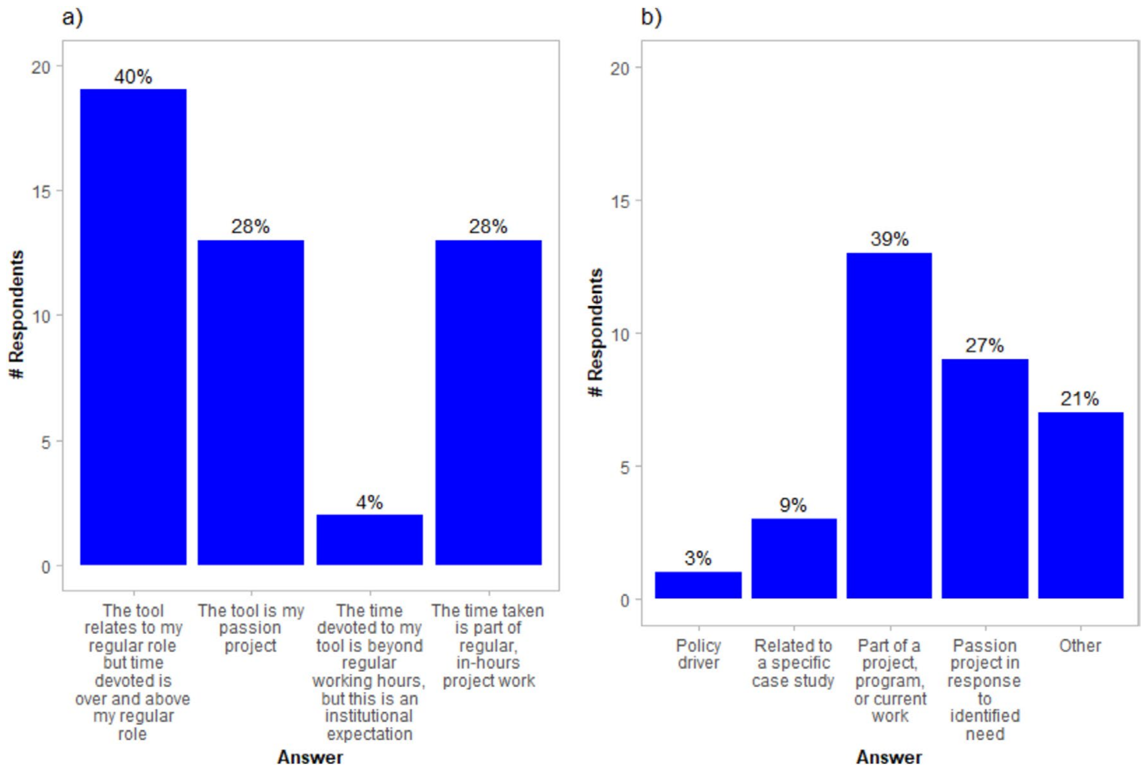

FIGURE 2 | Bar charts summarising frequencies of responses to the survey questions (a), "Do you view the time taken to develop, update and maintain your tool as being over and above your regular role" e.g., on your own time, or as a "passion project"? "Or is this time commitment an institutional expectation? (choose as many as are applicable)", and (b) "What was the origin (original application/need) and motivation of your tool"?

4.2.5 | Maintenance

Funding sources were unchanged from upscaling to maintenance for 88% of respondents. However, ongoing funding remained a major issue, with 42% self-funding maintenance (up from 34% at upscaling), with three respondents having lost funding from external clients/funding agencies, employers, or non-government agencies.

Concomitantly, 30% of respondents indicated their employer does not formally recognise and fund ongoing tool maintenance as core work, with a further 48% saying this occurs only to some extent (Figure 1). Of those 42% self-funding maintenance 43% (six respondents) stated their institution does not formally recognise tool maintenance as core work, while another 43% (six) said it does to some extent. The remaining two stated their institution does fund and recognise tool maintenance as core work, yet they were still self-funding this stage.

Developers at institutions providing no formal recognition were more likely to self-fund maintenance, though this difference was not statistically significant, likely due to small sample size (60% vs. 35%–33% at institutions with recognition; Fisher's exact test, $p = 0.53$) and reported higher levels of work outside regular hours, though these again were not statistically significant (Spearman's rho = −0.23, $p = 0.20$) (Table 2).

Time spent per month on tool updating and maintenance varied substantially. Of the 25 respondents providing estimates, the range spanned zero to 160 h per month, with almost half spending 10–40 h monthly (Figure 1). 20% reported 40+ hours monthly, while 20% noted sporadic work varying with funding and project needs. Interestingly, tools with full institutional recognition required more maintenance hours (median 25 h/month) than those with no recognition (median 10 h/month), possibly reflecting that better-supported tools are more actively maintained and developed (though the difference was not statistically significant—Kruskal–Wallis test, $p = 0.71$) (Table 2). Fewer monthly maintenance hours (median 10 h/month) were associated with self-funded than non-self-funded tools (median 24 h; Mann–Whitney U test, $p = 0.006$) (Figure 1 and Table 2).

Only 16% said no updating/maintenance work occurred outside regular hours, with 25% saying most or all updating and maintenance occurs outside hours (Figure 3b). The most common justifications were lack of time (25%) and that the tool was a passion project (20%) (Figure 3c). Of those citing "other" (18%), 70% cited lack of or unrealistically low funding.

Critically, none of the respondents spending out-of-hours time on updating/maintenance felt definitively that their tool would be as effective without this investment. 57% stated their tool would not be as effective without out-of-hours time, while the remainder felt it would "maybe" still be effective. Five of the eight "maybe" responders (63%) had indicated that they spend only "a little" time outside regular hours on updating/maintenance.

Based on prose responses, 76% indicated they had active ongoing personal interest in their tools, driven by continued use, feature

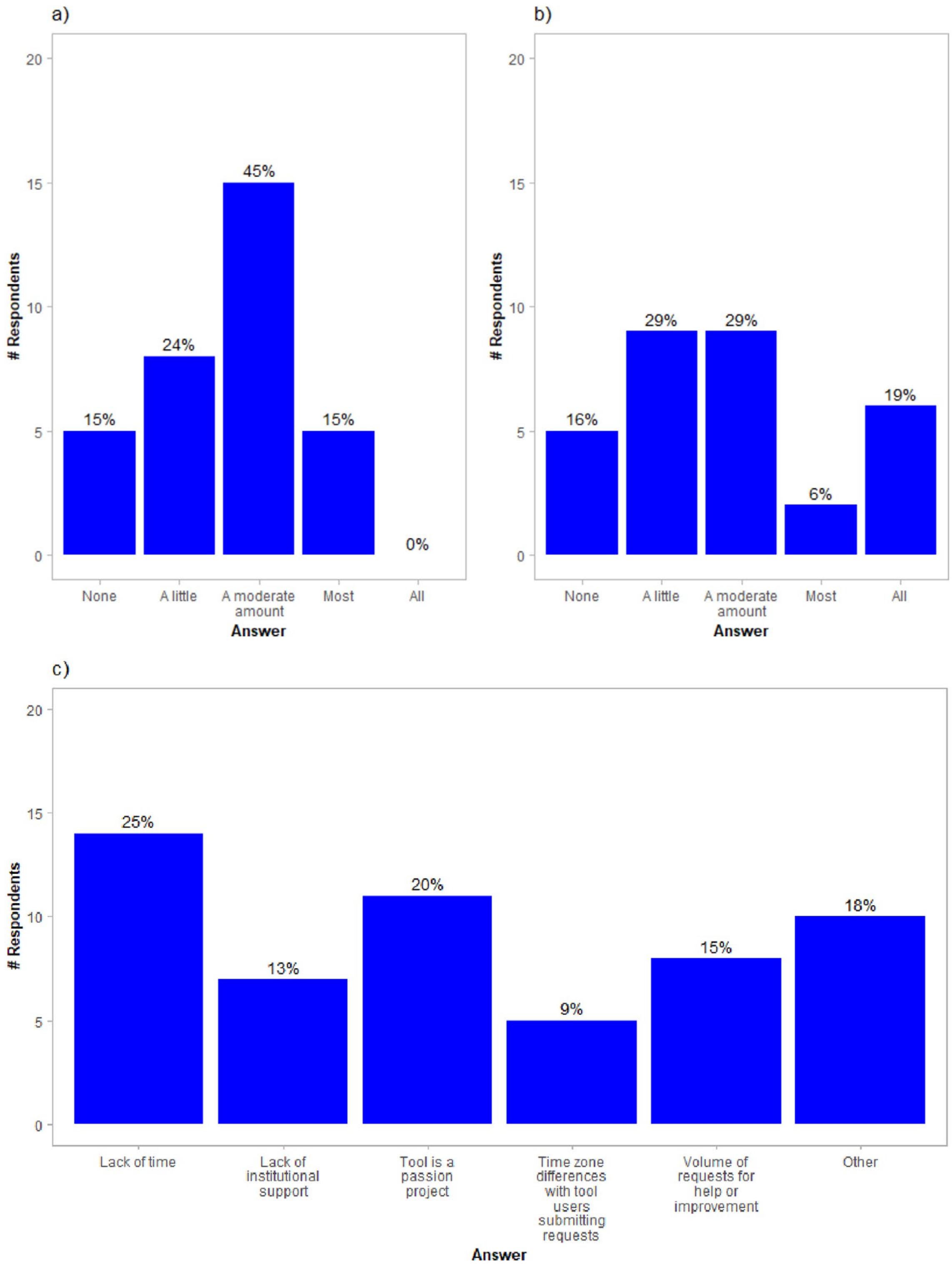

FIGURE 3 | Bar charts summarising frequencies of responses to the survey questions (a), "How much of the *transition* to a generalisable tool was undertaken outside of regular working hours?"; (b) "How much of tool *maintenance* is undertaken outside of regular working hours?" and (c) "Why is tool updating and maintenance undertaken outside of regular working hours"?

additions, or application to new contexts. Four respondents indicated they had moved on or lost interest, while five noted challenges maintaining involvement, often due to job changes or time constraints. Personal passion was explicitly mentioned by four respondents as a driving force, with three specifically noting continued involvement despite changing employers or roles, on their own time.

4.2.6 | Succession

61% of respondents indicated that they or someone else was involved in trainings to support succession. Of these, 65% directly led trainings, while 35% indicated this was done by others. However, only 12 respondents (36%) indicated they had or that there was a succession plan for their tool (Figure 1). Employer support levels did not differ significantly between tools with and without succession plans at any lifecycle stage (Kruskal–Wallis test, all $p > 0.20$) (Table 2).

4.2.7 | Obsolescence

30% of respondents (10) felt their tool was facing obsolescence (Figure 1). Of these, 60% (six) cited evolutionary reasons: tools at risk of being superseded (two), content replicated in more prominent tools (one), or improved tools being developed (three). Two cited an inability to maintain their tool long-term.

However, only 6% (two respondents) indicated "probably yes" when asked about loss of user interest, and of these, only one felt their tool was facing obsolescence. Almost 64% felt there "definitely" (39%) or "probably" (24%) had been no loss of interest by users, while the remaining 30% were unsure.

Obsolescence rates did not differ significantly by self-funding status (Fisher's exact test, $p = 0.71$), succession planning ($p = 1.0$), employer support levels at any lifecycle stage (Kruskal-Wallis tests, all $p > 0.23$), outside hours work at maintenance (Mann–Whitney U test, $p = 0.43$), or tool type (Fisher's exact test, $p = 0.67$) (Figure 1 and Table 2).

4.3 | Changes Across the Development and Use Lifecycle

Figure 4 presents three key quantitative patterns across the lifecycle: a funding "scissor effect" where institutional support declines as self-funding increases (a), stable employer recognition despite this funding shift (b), and persistent out-of-hours work across stages (c). These patterns are discussed in detail below.

For the 32 tools with complete data, the percentage of self-funded tools increased significantly from transition (18%) to maintenance (42%; McNemar's test, $\chi^2 = 6.13$, df = 1, $p = 0.013$) (Table 2), while institutionally-funded tools declined from 58% to 39% over the same period (Figures 1, 4a, 5). The decline in institutional funding as self-funding increased was most pronounced between the transition and maintenance phases.

Examining individual tool trajectories, 31% of tools (10 of 32) shifted from institutional to self-funding between inception and maintenance, while only 19% (6 of 32) experienced the opposite (Figure 6). In contrast, employer support remained relatively stable across lifecycle stages (mean support level ranging from 1.88 to 2.03 on a 1–3 scale; Friedman test, $\chi^2 = 1.83$, df = 3, $p = 0.61$) (Figures 1 and 4b; Table 2), where support represents employer interest/involvement at inception through upscaling stages, and formal recognition and funding of maintenance as core work at the maintenance stage. This stability suggests that the shift to self-funding reflects systematic underfunding of maintenance rather than declining institutional interest.

In contrast to the systematic shift in funding sources, work outside of standard hours remained consistently high across lifecycle stages (~60% of developers reported working a moderate amount or more outside regular hours during transition, compared to 55% during maintenance) (Figures 1 and 4c), with no significant difference between the transition and maintenance phases (Wilcoxon signed-rank test, $p = 0.61$) (Table 2). This persistence of voluntary effort despite declining institutional funding underscores the reliance on developer commitment to maintain tool viability.

Self-funded tools were significantly less likely to have succession plans (14%) compared to non-self-funded tools (53%; Fisher's exact test, $p = 0.033$) (Figures 1 and 7; Table 2). Higher employer support was associated with reduced outside hours work at the transition and maintenance stages: Spearman correlations approached significance (transition: rho = −0.30, $p = 0.094$; maintenance: rho = −0.28, $p = 0.12$), as did ordinal regression: high employer support at transition (coefficient = −1.58, $p = 0.095$) and full employer recognition at maintenance (coefficient = −1.67, $p = 0.090$) both showed lower out-of-hours work levels compared to those under low/no employer support (Table 2). However, the modest strength of these relationships indicates that substantial out-of-hours effort persists despite full employer recognition.

5 | Discussion

5.1 | Adjusting Expectations

Freely available online software tools are a valuable contribution to fisheries stock assessment science. We raised five expectations that we confront with our empirical findings.

First, we expected tools to follow a lifecycle of origin, transition, upscaling, maintenance and succession phases, with each having associated costs (Figure 1). The transition phase revealed departures from traditional software development lifecycles. Software development theory suggests structured progression through requirements, planning, design, and implementation (Hossain 2023; Boehm 1988; Royce 1970), but our results show a more organic process driven by practical needs and opportunities.

The finding that 45% of developers had full funding at conception, while 30% had none, indicates that successful tools can emerge regardless of initial financial support—a pattern more reminiscent of technology start-ups than traditional scientific software development (Ries 2011; Bhide 1992). This likely reflects developer motivation, faith in their tool's necessity, and commitment to improving stock assessment science.

Next, we expected that tools originated from specific applications before a broader need became apparent. Indeed, over a third of tools emerged from existing project needs, but 27% began as a personal initiative or passion project—although these may also have been addressing identified needs. Three respondents indicated that their tools were developed and applied to address general needs from the point of inception.

We postulated that tool lifecycle costs are borne, at least in part, by the developer, and at least some of a tool's development and maintenance is undertaken at the developer's behest. Developers faced major constraints beyond the initial development of their tools, self-funding upscaling in one-third of cases.

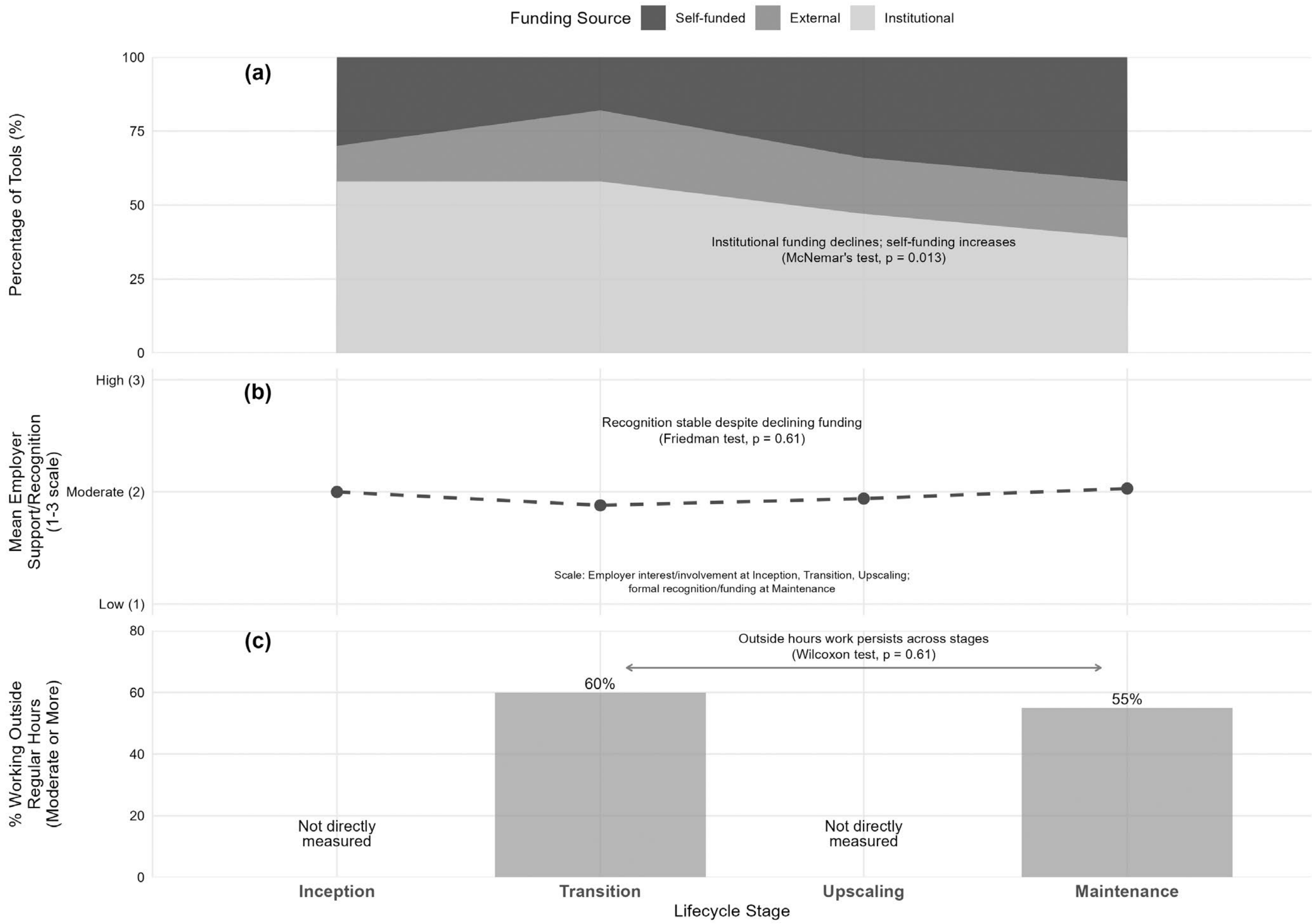

FIGURE 4 | Quantitative patterns across tool lifecycle stages. (a) Funding source composition across lifecycle stages. Institutional funding (light grey) declined from 58% to 39% between transition and maintenance, while self-funding (dark grey) increased from 18% to 42% (McNemar's test, $p = 0.013$). External funding (medium grey) remained relatively stable. (b) Mean employer support/recognition (1–3 scale) across lifecycle stages. The scale represents employer interest/involvement at Inception through upscaling stages, and formal recognition and funding of maintenance as core work at the maintenance stage (Friedman test, $p = 0.61$). (c) Percentage of developers working outside regular hours (moderate amounts or more) at transition and maintenance stages (outside hours work was not directly measured at Inception and Upscaling stages) (Wilcoxon signed-rank test comparing transition vs. maintenance, $p = 0.61$).

Almost one-third of respondents to our survey felt some lack of employer prioritisation or resistance. Time and funding were major upscaling challenges for 48% of respondents.

The maintenance phase emerged as the most challenging. Developers devoted ~10–40 h per month to tool maintenance. However, that 42% of developers self-fund maintenance, up from 33% during upscaling (and compared to 30% of tools being unfunded at point of inception), suggests a systematic undervaluation of tool maintenance in institutional planning.

60% of respondents undertook at least some of the work outside regular hours during transition, and 84% undertook at least a little (and 55% at least a moderate amount) of the work outside regular hours during the maintenance phase. None felt their tools would be as effective without out-of-hours commitment, highlighting a concerning reliance on volunteerism.

We expected a risk point to be misalignment in tool expectations between developers and employers/funders. Survey results showed a lack of formal support, or reduction in formal support beyond a tool's initial development, often leading to increased personal time investment. Institutional funding frequently and progressively diminished during the upscaling and maintenance phases, despite the ongoing importance of the tool. These were the strongest emerging themes and will be discussed in more detail below.

There is a low rate of succession planning (36% of respondents) despite high training involvement (61% of respondents), suggesting a disconnect between short-term operational needs and long-term sustainability. This was particularly acute for self-funded tools, which were less likely to have succession plans (14%) compared to institutionally funded tools (53%; Figure 1). This gap represents a key risk point and raises important questions about the long-term viability of the current development model. Tools generally either continue to be updated or become superseded or obsolescent when they are not updated. Regardless, developers indicated almost no perceived loss of interest in their tool by users. The higher maintenance hours associated with tools that have succession plans may suggest that these tools are better supported and are more actively developed/maintained (not abandoned).

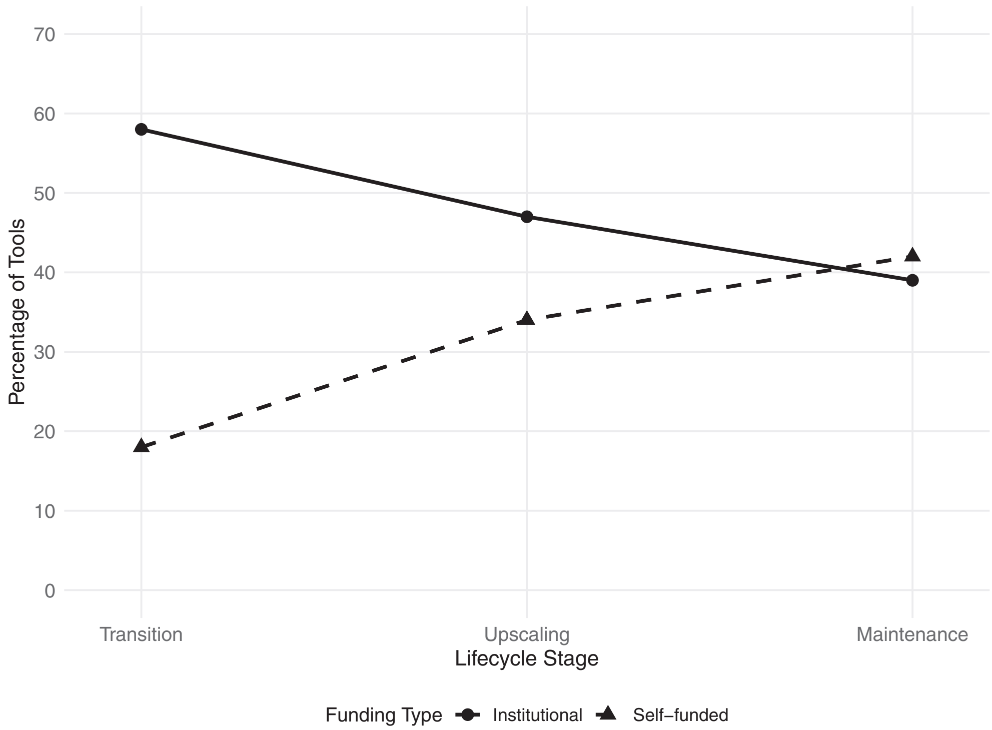

FIGURE 5 | Percentage of tools with institutional (solid line; circles) versus self-funding (dashed line; triangles) across the lifecycle stages of transition through to maintenance ($n = 32$). Self-funding increases significantly from transition (18%) to maintenance (42%; McNemar's test: $\chi^2 = 6.13$, $p = 0.013$), while institutional funding declines from 58% to 39% over the same period.

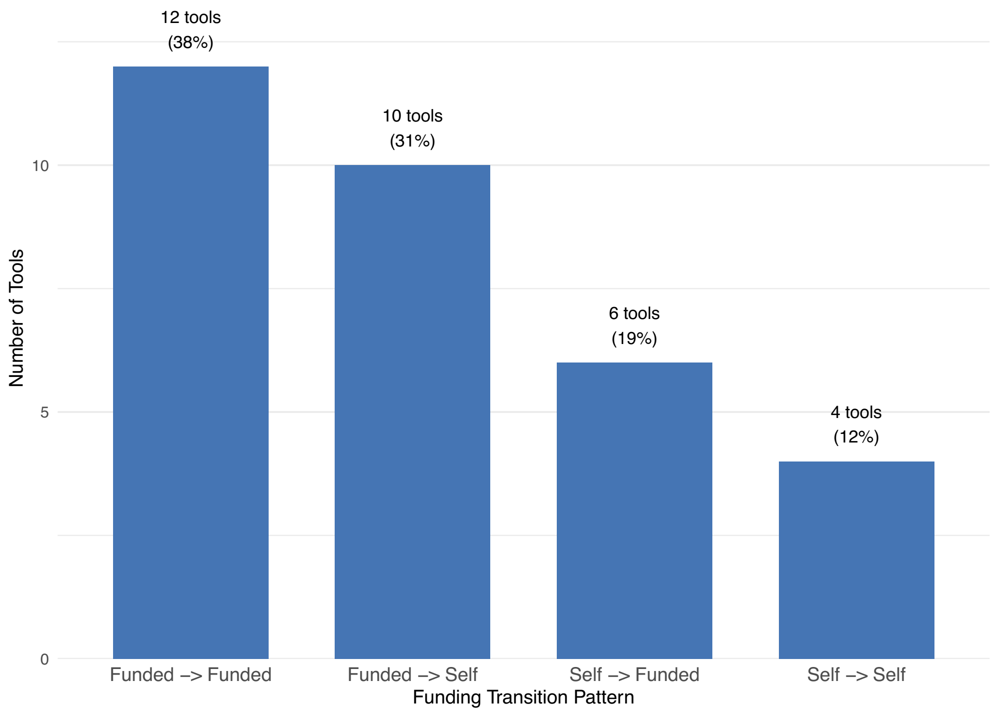

FIGURE 6 | Tool funding transitions from inception to maintenance ($n = 32$).

Finally, we expected developers to lag on tool maintenance when their tool lacks support in terms of ongoing funding, personnel/capacity or a community of developers. Our expectation that tools "in the margins" with regard to funding and in-kind agency support do not survive was only partially supported: tools survive despite, not because of, such institutional support. Despite moderate employer support, developers work beyond regular hours, particularly during upscaling and maintenance. Many tools persist despite limited employer support (in terms of recognising the value of the tool and the

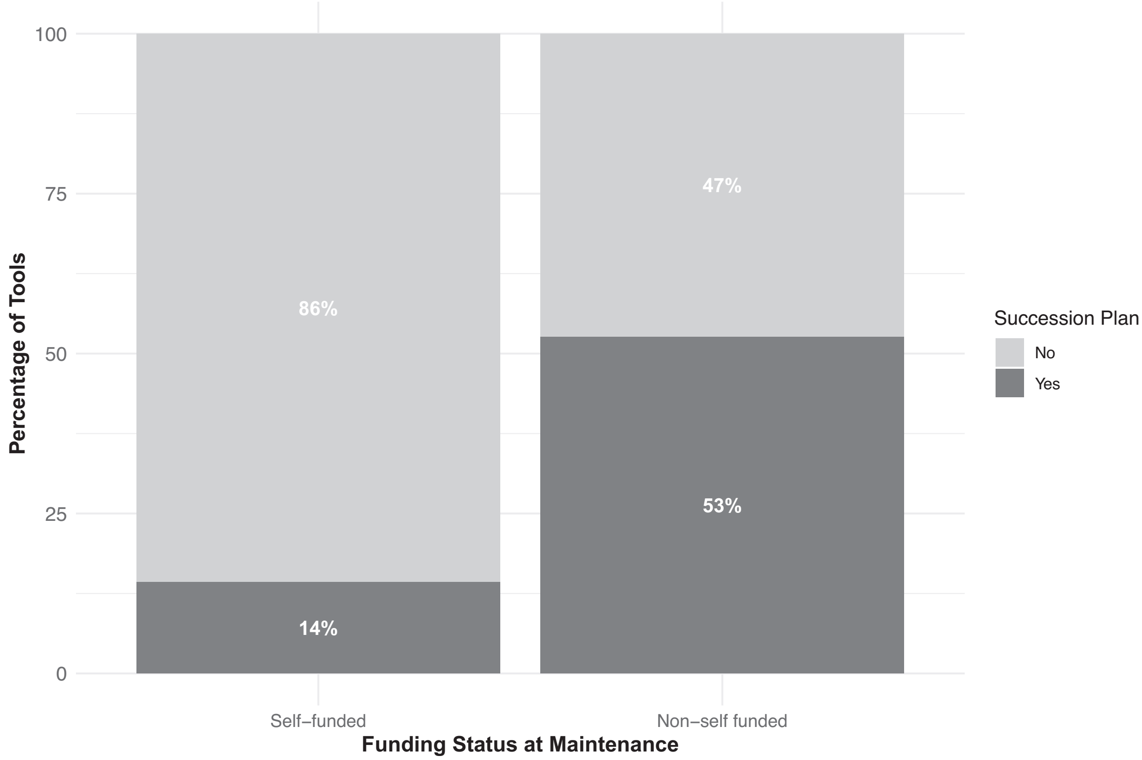

FIGURE 7 | Presence of a succession plan, by whether the tool is self-funded at the maintenance stage.

need for dedicated time to continue its development and maintenance), primarily through substantial voluntary effort from dedicated developers.

These issues are likely to be relevant to applied science tools generally, where the developers are practitioners as opposed to formally trained software developers.

5.2 | Are Tools Developed and Maintained Through Acts of Volunteerism?

Our findings suggest tool development relies heavily on what the volunteerism literature terms 'sustained volunteering' – a long-term personal commitment beyond institutional support (Hustinx et al. 2010; Snyder and Omoto 2008; Penner 2002; Clary et al. 1998). That 84% of respondents undertake at least some maintenance outside regular hours parallels both volunteerism theory and business development, where personal investment precedes institutional support (Dutta-Bergman 2004; Winborg and Landström 2001; Wilson 2000), consistent with Handy et al.'s (2000) definition of volunteerism as an individual incurring relatively higher net cost.

Tool development volunteerism aligns with Clary et al.'s (1998) motivating factors of values (concern for others), understanding (exercising new or otherwise unpractised knowledge and skills), and career benefits. It also fits the economic 'public goods' model, where individuals donate time to increase provision of valued public services (Hustinx et al. 2010).

It is important, however, to consider why fisheries stock assessment tool development and maintenance become acts of what we are terming "volunteerism". Volunteerism is a social construct with multiple definitions (Hustinx et al. 2010). People performing identical tasks willingly and without remuneration may not define this as volunteering, but rather as public service or role expectations. Tool development may represent 'situational extension'—going beyond expected duties because stock assessment scientists are frequently busy and work in fields without strict time boundaries. Anecdotally, many developers care deeply about conservation, sustaining fisheries, and supporting their scientific community. Developers create software products requiring care and time to nurture and maintain, but the associated unpredictable required time makes formal workplan integration difficult.

Affording developers more official time raises the question, "Where does this end?", especially considering developers may create multiple tools, and that most are scientists with other duties. Scientists in government agencies or universities face finite resources and few new hires, preventing handover of tools. Ultimately, tool developers must invest extra time to continue development and maintenance—often an unspoken institutional expectation. Private consultants find it difficult to contribute to tool development and maintenance unless they absorb costs within funded projects or donate time freely.

Tool developers generally do not profit financially. U.S. government workers particularly attempt to ensure freely available products. Objectives centre on education, food security, regional employment, and sustainable resources—valuable goals, but tools generally are not sold. Our survey demonstrates that user-free tools impose costs on developers. At some point, tool support becomes the developer's own. We advocate for better support of tools designed to help, as opposed to generate revenue.

Whether 'volunteerism' would persist with better support is uncertain. Wilson (2000) points out that, as the level of respect afforded to one's job increases, so does the likelihood of volunteering, and that it is probable that professional-level people who volunteer get more intrinsic rewards from their work, building an attachment to work and work-like activities that easily translates into volunteerism (Herzog and Morgan 1993). Certain volunteerism aspects should persist, particularly user contributions. Even with full-time formal allocation to tool development, user contributions remain valuable, freeing developers for other advancements.

5.3 | Maintaining Tools Requires Sustained Investment

Software development theory cautions that deferring maintenance leads to technical debt—future costs incurred when organisations defer maintenance, leading to compounding problems and substantially higher long-term costs (Kruchten et al. 2012; Brown et al. 2010; Sneed and Brössler 2003; Banker and Slaughter 2000). Business theory on operational sustainability similarly stresses that long-term viability requires dedicated resources and clear ownership structures (Drucker 1999; Porter and Kramer 2006; Teece 2007).

Our survey revealed that tools were significantly more likely to be self-funded during maintenance than during transition (McNemar's test; $p = 0.013$), despite stable levels of employer recognition across stages (Figure 4b). This paradox—where employers recognise the value of tools but progressively withdraw funding support—suggests institutional acknowledgment without resource commitment. This shift in funding (recalling 45% of tools had full funding at inception, but 42% became self-funded during maintenance) creates a "scissor effect" (Figures 4a, 5), empirically demonstrating the undervaluation of maintenance work identified in the software development literature (Kruchten et al. 2012; Brown et al. 2010).

Science's reward structures typically emphasise novel contributions. Tool maintenance work, while valuable, is often not recognised in the same way as developing new methods or publishing research. Yet maintenance occurs regardless, often borne by developers' voluntary effort. Funding bodies and institutions should recognise tool maintenance as a worthwhile investment; indeed, supporting existing successful tools often costs less than developing new ones.

Tool adoption followed patterns described in technology diffusion theory (Rogers 2003; Chandler 1990; Penrose 1959), with uptake ranging from under 1 year to over 5 years. Workshops, conferences, and teaching proved most effective for dissemination, consistent with network-based community building (Parker and van Alstyne 2005; McAlexander et al. 2002; Katz and Shapiro 1985). Yet 54% of developers do not actively promote their tools, indicating a missed opportunity for broader impact. Tool developers could prioritise active promotion and dissemination, provided they have time to support a broader user base and can do so responsibly to encourage critical use. Overselling while downplaying limitations and risks should be avoided.

Stock assessment scientists are not typically trained software developers. Partnerships between scientists and professional developers could improve development efficiency and tool quality. Scientists would develop concepts and prototypes; developers would handle generalisation and scaling. This handover requires clear communication and relinquishing control—transitions that parallel small business expansion and carry similar risks (Combs and Ketchen Jr 2003; Stanworth et al. 1998). Alternatively, investments in raising scientists' coding skills, bringing in developers early for guidance, or cautious use of AI coding assistance could build capacity.

Regardless, employers and universities should be encouraged to provide robust development infrastructures. Software development is increasingly important in fisheries science, requiring support in both professional and educational career phases. Support and training vary considerably in university curricula, with many students learning code as a class byproduct or finding informal learning opportunities voluntarily. Fredstone and Lowndes (2024) emphasise that upskilling requires approved paid time and is a continual process.

5.4 | Survey Considerations

Our survey could have benefited from a more flexible, inclusive design. The survey assumed that tool developers worked as part of a larger organisation/institution and remained with this organisation throughout the tool's evolution. Some respondents were self-employed or government contractors. One respondent had left an employer with no interest in maintaining their tool after departure, deliberately made their tool open source, and subsequently developed an improved version. Such situations were not directly covered by the survey.

We assumed original tool developers were scientists rather than formally trained software developers, and that if software developers were brought in, this occurred partway through development. While we asked at two separate stages whether people brought in formal programmers, we failed to ask whether this occurred during initial development.

One respondent stated that the survey seemed designed for more mature/complete tools. Tools were selected from the Australian Stock Assessment Toolbox (Dichmont et al. 2021), with additional key tools added based on the authors' collective knowledge, so only well-known tools were included. While this creates selection bias—less widely known tools may have been missed—our intention was not to be exhaustive but to obtain a reasonable subsample of available tools to test our a priori expectations.

We assumed that lack of money, time, resources, and programming skills were key constraints limiting widespread tool uptake. We failed to consider "soft skills" that help tools become widely used, including quality documentation, teaching skills for training users, and creating welcoming tool communities (like GitHub) that provide ongoing support for questions and contributions. Some newer tools with quality documentation and active user communities are more widely used than longer-established tools.

5.5 | Facilitating Sustainable Tool Development

Survey responses and specific feedback highlight several factors that could make tool development and maintenance more sustainable. Institutional philosophy and flexibility are critical enablers. Employers that allow discretionary time and accept innovative thinking—even with its inherent risk of failure—better nurture tool development. Developer motivation also plays a crucial role: some researchers view tool development as an essential service to the scientific community, while others prefer to publish methods and leave implementation to others. Funding structures typically favour new tool development over maintenance of existing tools, creating perverse incentives to develop new tools rather than maintain and improve existing ones.

The current global reliance on personal time investment for free scientific tools is likely unsustainable. While this volunteerism has enabled freely available fishery stock assessment tools to proliferate and sustain, it creates critical infrastructure vulnerabilities and may discourage potential developers who cannot commit substantial personal time. Another caution is the possibility of maladaptive responses to volunteering, such as burnout (Hustinx et al. 2010).

A more sustainable model requires explicit institutional recognition and support for both development and maintenance of freely available stock assessment software tools. This includes formal recognition and funding mechanisms for tool maintenance within research institutions, with dedicated funding streams for long-term maintenance—challenging because ongoing maintenance is generally not deemed attractive by some funders, reinforcing the need for institutions to support maintenance as core business. Development and maintenance should be better integrated into core institutional workflows. Professional development pathways for scientific software developers should be established (organisations like The Society of Research Software Engineering https://society-rse.org/ work on such issues), or institutions should facilitate pairing with professional software developers—noting that most fishery stock assessment tool developers are not primarily software developers. Finally, structured succession planning frameworks should be developed.

5.6 | Conclusion

Free fisheries software tools endure despite lack of institutional support, only because developers invest significant personal time. Key risk points are insufficient institutional support in terms of time and money. Developers work beyond expectations to maintain their tools, even when unsupported. As such, tools largely represent acts of volunteerism. While this demonstrates admirable commitment to scientific advancement, and while volunteerism may be effective short-term, it does not represent a sustainable model for maintaining freely available fishery stock assessment software tools. Future efforts should better integrate tool development and maintenance into institutional frameworks while preserving the innovation and dedication that characterise the current system. We advocate for better support of tool developers, and for tools—particularly their ongoing maintenance—to become internalised and owned as part of core business.

Acknowledgements

We wish to thank two anonymous reviewers and the editor for comments on an earlier version of this manuscript. This work was partly funded by the Fisheries Research and Development Corporation (Project No. 2021-002), Australia, with additional in-kind contributions from the CSIRO and Cathy Dichmont Consulting. In-kind contributions were also provided by the "free tool" developers and teams, all of whom we sincerely thank for so willingly engaging with our survey, providing additional feedback, and supporting the paper.

Funding

This work was supported by Fisheries Research and Development Corporation, 2021-002. Commonwealth Scientific and Industrial Research Organisation. Cathy Dichmont Consulting Pty. Ltd.

Conflicts of Interest

J.C., C.D., N.D., and A.P. were among the tool developers who undertook the survey.

Data Availability Statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

Adams, G. D., K. K. Holsman, S. J. Barbeaux, et al. 2022. "An Ensemble Approach to Understand Predation Mortality for Groundfish in the Gulf of Alaska." *Fisheries Research* 251: 106303. https://doi.org/10.1016/j.fishres.2022.106303.

Anderson, C., C. C. Monnahan, K. F. Johnson, K. Ono, and J. L. Valero. 2014. "ss3sim: An R Package for Fisheries Stock Assessment Simulation With Stock Synthesis." *PLoS One* 9, no. 4: e92725. https://doi.org/10.1371/journal.pone.0092725.

Anderson, S. C., E. J. Ward, P. A. English, L. A. K. Barnett, and J. T. Thorson. 2024. "sdmTMB: An R Package for Fast, Flexible, and User-Friendly Generalized Linear Mixed Effects Models With Spatial and Spatiotemporal Random Fields." *Journal of Statistical Software* 115, no. 2: 1–46. https://doi.org/10.18637/jss.v115.i02.

Banker, R. D., and S. A. Slaughter. 2000. "The Moderating Effects of Structure on Volatility and Complexity in Software Enhancement." *Information Systems Research* 11, no. 3: 219–240. https://doi.org/10.1287/isre.11.3.219.12209.

Begley, J. 2006. Gadget User Guide. https://hafro.github.io/gadget2/docs/userguide/.

Bhide, A. 1992. "Bootstrap Finance: The Art of Start-Ups." *Harvard Business Review* 70, no. 6: 109–117.

Boehm, B. W. 1988. "A Spiral Model of Software Development and Enhancement." *Computer* 21, no. 5: 61–72. https://doi.org/10.1109/2.59.

Brown, N., Y. Cai, Y. Guo, et al. 2010. "Managing Technical Debt in Software-Reliant Systems." *Workshop on Future of Software Engineering Research*: 47–52. https://doi.org/10.1145/1882362.1882373.

Carvalho, F., H. Winker, D. Courtney, et al. 2021. "A Cookbook for Using Model Diagnostics in Integrated Stock Assessments." *Fisheries Research* 240: 105959. https://doi.org/10.1016/j.fishres.2021.105959.

Casal2 Development Team. 2023. *Casal2 User Manual for Age-Based Models, v23.09 (2023-09-16)*, 268. National Institute of Water & Atmospheric Research Ltd. NIWA Technical Report 139. https://raw.githubusercontent.com/NIWAFisheriesModelling/CASAL2/master/Documentation/UserManual/CASAL2_Age.pdf.

Chandler, A. D. 1990. *Scale and Scope: The Dynamics of Industrial Capitalism*. Harvard University Press.

Churchill, N. C., and V. L. Lewis. 1983. "The Five Stages of Small Business Growth." *Harvard Business Review* 61, no. 3: 30–50.

Clary, E. G., M. Snyder, R. D. Ridge, et al. 1998. "Understanding and Assessing the Motivations of Volunteers: A Functional Approach." *Journal of Personality and Social Psychology* 74, no. 6: 1516–1530. https://doi.org/10.1037/0022-3514.74.6.1516.

Combs, J. G., and D. J. Ketchen Jr. 2003. "Why Do Firms Use Franchising as an Entrepreneurial Strategy?: A Meta-Analysis." *Journal of Management* 29, no. 3: 443–465. https://doi.org/10.1016/S0149-2063_03_00019-9.

Cope, J. M. 2020. The Stock Synthesis Data-Limited Tool (SS-DL Tool). https://github.com/shcaba/SS-DL-tool#the-stock-synthesis-data-limited-tool-ss-dl-tool.

Cope, J. M. 2024. "The Good Practices of Practicable Alchemy in the Stock Assessment Continuum: Fundamentals and Principles of Analytical Methods to Support Science-Based Fisheries Management Under Data and Resource Limitations." *Fisheries Research* 270: 106859. https://doi.org/10.1016/j.fishres.2023.106859.

Dichmont, C. M., R. A. Deng, N. Dowling, and A. E. Punt. 2021. "Collating Stock Assessment Packages to Improve Stock Assessments." *Fisheries Research* 236: 105884. https://doi.org/10.1016/j.fishres.2020.105844.

Dichmont, C. M., R. A. Deng, A. E. Punt, et al. 2016. "A Review of Stock Assessment Packages in the United States." *Fisheries Research* 183: 447–460. https://doi.org/10.1016/j.fishres.2016.07.001.

Doering, K. L., N. R. Vaughan, J. F. Walter, et al. 2023. "SSMSE: An R Package for Management Strategy Evaluation With Stock Synthesis Operating Models." *Journal of Open Source Software* 8, no. 90: 4937. https://doi.org/10.21105/joss.04937.

Dowling, N. A., J. R. Wilson, J. M. Cope, et al. 2023. "The FishPath Approach for Fisheries Management in a Data- and Capacity Limited World." *Fish and Fisheries* 24, no. 2: 212–230. https://doi.org/10.1111/faf.12721.

Drucker, P. F. 1999. *Management Challenges for the 21st Century*. Harper Business.

Dutta-Bergman, M. J. 2004. "Describing Volunteerism: The Theory of Unified Responsibility." *Journal of Public Relations Research* 16, no. 4: 353–369. https://doi.org/10.1207/s1532754xjprr1604_2.

Fournier, D. A., J. Hampton, and J. R. Sibert. 1998. "MULTIFAN-CL: A Length-Based, Age-Structured Model for Fisheries Stock Assessment, With Application to South Pacific Albacore, *Thunnus alalunga*." *Canadian Journal of Fisheries and Aquatic Sciences* 55, no. 9: 2105–2116. https://doi.org/10.1139/f98-100.

Fredstone, A. L., and J. S. Lowndes. 2024. "Welcoming More Participation in Open Data Science for the Oceans." *Annual Review of Marine Science* 16: 537–549. https://doi.org/10.1146/annurev-marine-041723-094741.

Fulton, E. A., J. S. Link, I. C. Kaplan, et al. 2011. "Lessons in Modelling and Management of Marine Ecosystems: The Atlantis Experience." *Fish and Fisheries* 12: 171–188. https://doi.org/10.1111/j.1467-2979.2011.00412.x.

Haddon, M. 2021. *Using R for Modelling and Quantitative Methods in Fisheries*, 337. CRC/Chapman & Hall. https://haddonm.github.io/URMQMF/.

Haddon, M., P. Burch, N. Dowling, and L. R. Little. 2019. *Reducing the Number of Undefined Species in Future Status of Australian Fish Stocks Reports: Phase Two - Training in the Assessment of Data-Poor Stocks*, 125. FRDC Final Report 2017/102. CSIRO Oceans and Atmosphere and Fisheries Research Development Corporation.

Handy, F., R. A. Cnaan, J. L. Brudney, U. Ascoli, L. C. M. P. Meijs, and S. Ranade. 2000. "Public Perception of "Who Is a Volunteer": An Examination of the Net-Cost Approach From a Cross-Cultural Perspective." *Voluntas: International Journal of Voluntary and Nonprofit Organizations* 11, no. 1: 45–65. https://doi.org/10.1023/A:1008903032393.

Herzog, A., and J. Morgan. 1993. "Formal Volunteer Work Among Older Americans." In *Achieving a Productive Aging Society*, edited by S. Bass, F. Caro, and Y. Chen, 119–142. Auburn House.

Holsman, K. K., J. Ianelli, K. Aydin, A. E. Punt, and E. A. Moffitt. 2016. "A Comparison of Fisheries Biological Reference Points Estimated From Temperature-Specific Multispecies and Single Species Stock Assessment Models." *Deep Sea Research. Part II, Topical Studies in Oceanography* 134: 360–378. https://doi.org/10.1016/j.dsr2.2015.08.001.

Hordyk, A. R., T. R. Carruthers, and Q. Huynh. 2022. openMSE: Open-Source Libraries for Management Strategy Evaluation. https://openmse.com/.

Hossain, M. I. 2023. "Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management." *International Journal for Multidisciplinary Research* 5, no. 5: 1–36.

Hustinx, L., R. A. Cnaan, and F. Handy. 2010. "Navigating Theories of Volunteering: A Hybrid Map for a Complex Phenomenon." *Journal for the Theory of Social Behaviour* 40, no. 4: 410–434. https://doi.org/10.1111/j.1468-5914.2010.00439.x.

Ianelli, J. 2022. AMAK. GitHub. Accessed 15th February, 2026. https://github.com/afsc-assessments/AMAK/blob/master/docs/AMAK%20Documentation.pdf.

Jardim, E., C. P. Millar, I. Mosqueira, et al. 2014. "What if Stock Assessment Is as Simple as a Linear Model? The a4a Initiative." *ICES Journal of Marine Science* 72: 232–236. https://doi.org/10.1093/icesjms/fsu050.

Katz, M. L., and C. Shapiro. 1985. "Network Externalities, Competition, and Compatibility." *American Economic Review* 75, no. 3: 424–440. https://www.jstor.org/stable/1814809.

Kell, L. T., I. Mosqueira, P. Grosjean, et al. 2007. "FLR: An Open-Source Framework for the Evaluation and Development of Management Strategies." *ICES Journal of Marine Science* 64: 640–646. https://doi.org/10.1093/icesjms/fsm012.

Kelly, D. 2015. "Scientific Software Development Viewed as Knowledge Acquisition: Towards Understanding the Development of Risk-Averse Scientific Software." *Journal of Systems and Software* 109: 50–61. https://doi.org/10.1016/j.jss.2015.07.027.

Kruchten, P. 2004. *The Rational Unified Process: An Introduction*. 3rd ed. Addison-Wesley Professional.

Kruchten, P., R. L. Nord, and I. Ozkaya. 2012. "Technical Debt: From Metaphor to Theory and Practice." *IEEE Software* 29, no. 6: 18–21.

Lentin, J., B. Elvarsson, and W. Butler. 2025. _gadget3: Globally-Applicable Area Disaggregated General Ecosystem Toolbox V3. R Package Version 0.13-0. https://doi.org/10.32614/CRAN.package.gadget3.

Lucey, S. M., S. K. Gaichas, and K. Y. Aydin. 2020. "Conducting Reproducible Ecosystem Modeling Using the Open Source Mass Balance Model Rpath." *Ecological Modelling* 427: 109057. https://doi.org/10.1016/j.ecolmodel.2020.109057.

McAlexander, J. H., J. W. Schouten, and H. F. Koenig. 2002. "Building Brand Community." *Journal of Marketing* 66, no. 1: 38–54. https://doi.org/10.1509/jmkg.66.1.38.18451.

Methot, R. D., Jr., and C. R. Wetzel. 2013. "Stock Synthesis: A Biological and Statistical Framework for Fish Stock Assessment and Fishery Management." *Fisheries Research* 142: 86–99. https://doi.org/10.1016/j.fishres.2012.10.012.

Mildenberger, T. K., M. H. Taylor, and M. Wolff. 2017. "TropFishR: An R Package for Fisheries Analysis With Length-Frequency Data." *Methods in Ecology and Evolution* 8, no. 11: 1520–1527. https://doi.org/10.1111/2041-210X.12791.

Nielsen, A., N. T. Hintzen, H. Mosegaard, V. Trijoulet, and C. W. Berg. 2021. "Multi-Fleet State-Space Assessment Model Strengthens Confidence in Single-Fleet SAM and Provides Fleet-Specific Forecast Options." *ICES Journal of Marine Science* 78, no. 6: 2043–20521. https://doi.org/10.1093/icesjms/fsab078.

Parker, G., and M. van Alstyne. 2005. "Two-Sided Network Effects: A Theory of Information Product Design." *Management Science* 51, no. 10: 1494–1504. https://doi.org/10.1287/mnsc.1050.0400.

Pedersen, M. W., and C. W. Berg. 2017. "A Stochastic Surplus Production Model in Continuous Time." *Fish and Fisheries* 18: 226–243. https://doi.org/10.1111/faf.12174.

Penner, L. A. 2002. "Dispositional and Organizational Influences on Sustained Volunteerism: An Interactionist Perspective." *Journal of Social Issues* 58, no. 3: 447–467. https://doi.org/10.1111/1540-4560.00270.

Penrose, E. T. 1959. *The Theory of the Growth of the Firm*. Oxford University Press.

Porter, M. E., and M. R. Kramer. 2006. "Strategy and Society: The Link Between Competitive Advantage and Corporate Social Responsibility." *Harvard Business Review* 84, no. 12: 78–92.

Prince, J. D. 2003. "The Barefoot Ecologist goes Fishing." *Fish and Fisheries* 4: 359–371. https://doi.org/10.1046/j.1467-2979.2003.00134.x.

Ries, E. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, 336. Crown Business.

Rogers, E. M. 2003. *Diffusion of Innovations*. 5th ed, 576. Simon and Schuster.

Royce, W. W. 1970. *Managing the Development of Large Software Systems: Concepts and Techniques*, 1–9. Proceedings of IEEE WESCON, Los Angeles. This paper was later republished in 1987 in the Proceedings of the 9th International Conference on Software Engineering (ICSE), IEEE Computer Society Press, pp. 328–338.

Rudd, M. B., and J. T. Thorson. 2017. "Accounting for Variable Recruitment and Fishing Mortality in Length-Based Stock Assessments for Data-Limited Fisheries." *Canadian Journal of Fisheries and Aquatic Sciences* 75, no. 7: 1019–1035. https://doi.org/10.1139/cjfas-2017-0143.

Schwaber, K., and J. Beedle. 2002. *Agile Software Development With Scrum*, 158. Prentice Hall.

Schweik, C. M., and R. English. 2007. *Tragedy of the FOSS Commons? Investigating the Institutional Designs of Free/Libre and Open Source Software Projects*. Vol. 14. National Center for Digital Government Working Paper Series. https://doi.org/10.5210/fm.v12i2.1619.

Scott, M., and R. Bruce. 1987. "Five Stages of Growth in Small Business." *Long Range Planning* 20, no. 3: 45–52. https://doi.org/10.1016/0024-6301(87)90071-9.

Sneed, H. M., and P. Brössler. 2003. *Critical Success Factors in Software Maintenance: A Case Study*, 190–198. International Conference on Software Maintenance. https://doi.org/10.1109/ICSM.2003.1235421.

Snyder, M., and A. M. Omoto. 2008. "Volunteerism: Social Issues Perspectives and Social Policy Implications." *Social Issues and Policy Review* 2, no. 1: 1–36. https://doi.org/10.1111/j.1751-2409.2008.00009.x.

Stanworth, J., D. Purdy, S. Price, and N. Zafiris. 1998. "Franchise Versus Conventional Small Business Failure Rates in the US and UK: More Similarities Than Differences." *International Small Business Journal* 16, no. 3: 56–69. https://doi.org/10.1177/0266242698163003.

Stefansson, G., and O. Palsson. 1998. "A Framework for Multispecies Modelling of Arcto-Boreal Systems." *Reviews in Fish Biology and Fisheries* 8: 101–104.

Stock, B. C., and T. J. Miller. 2021. "The Woods Hole Assessment Model (WHAM): A General State-Space Assessment Framework That Incorporates Time- and Age-Varying Processes via Random Effects and Links to Environmental Covariates." *Fisheries Research* 240: 105967. https://doi.org/10.1016/j.fishres.2021.105967.

Taylor, I. G., K. L. Doering, K. F. Johnson, C. R. Wetzel, and I. J. Stewart. 2021. "Beyond Visualizing Catch-At-Age Models: Lessons Learned From the r4ss Package About Software to Support Stock Assessments." *Fisheries Research* 239: 105924. https://doi.org/10.1016/j.fishres.2021.105924.

Teece, D. J. 2007. "Explicating Dynamic Capabilities: The Nature and Microfoundations of (Sustainable) Enterprise Performance." *Strategic Management Journal* 28, no. 13: 1319–1350. https://doi.org/10.1002/smj.640.

Thimbleby, H. 2024. "Improving Science That Uses Code." *Computer Journal* 67, no. 4: 1381–1404. https://doi.org/10.1093/comjnl/bxad067.

Thorson, J. T. 2019. "Guidance for Decisions Using the Vector Autoregressive Spatio-Temporal (VAST) Package in Stock, Ecosystem, Habitat and Climate Assessments." *Fisheries Research* 210: 143–161. https://doi.org/10.1016/j.fishres.2018.10.013.

Veron, M., and C. Szuwalski. 2024. gmr: A Generalized Size-Structured Model for Assessing Crustaceans in R. https://gmacs-project.github.io/gmr/.

Wilson, J. 2000. "Volunteering." *Annual Review of Sociology* 26: 215–240. https://doi.org/10.1146/annurev.soc.26.1.215.

Winborg, J., and H. Landström. 2001. "Financial Bootstrapping in Small Businesses: Examining Small Business Managers' Resource Acquisition Behaviors." *Journal of Business Venturing* 16, no. 3: 235–254. https://doi.org/10.1016/S0883-9026(99)00055-5.

Winker, H., F. Carvalho, and M. Kapur. 2018. "JABBA: Just Another Bayesian Biomass Assessment." *Fisheries Research* 204: 275–288. https://doi.org/10.1016/j.fishres.2018.03.010.

Þór Elvarsson, B., P. J. Woods, H. Björnsson, J. Lentin, and G. Thordarson. 2018. "Pushing the Limits of a Data Challenged Stock: A Size- and Age-Structured Assessment of Ling (*Molva molva*) in Icelandic Waters Using Gadget." *Fisheries Research* 207: 95–109. https://doi.org/10.1016/j.fishres.2018.06.005.

Supporting Information

Additional supporting information can be found online in the Supporting Information section. **Table S1:** Full questionnaire and summaries of responses. **Data S1:** A brief synthesis of the "free tool" packages considered.