A prototype software framework for transparent, reusable and updatable health economic models

Matthew P Hamilton[1,2,✉], Caroline X Gao[2,3,1], Glen Wiesner[4], Kate M Filia[2,3], Jana M Menssink[2,3], Petra Plencnerova[5], David Baker[2,3], Patrick D McGorry[2,3], Alexandra Parker[6], Jonathan Karnon[7], Sue M Cotton[2,3], and Cathrine Mihalopoulos[1]

Summary: Computers are essential to the work of health economists. However, the ethical dimensions of how health economic models are implemented computationally are poorly understood and inadequately fulfilled. As tools to facilitate collaborative and efficient code development, software frameworks have a potential role to play in reducing barriers to ethical modelling practice. We propose six criteria for assessing ethical implementation of computational health economic models (CHEMs) – two each for the three domains of transparency, reusability and updatability. To facilitate the implementation of CHEMs that meet these criteria, we developed a novel prototype software framework in the open source programming language R. The framework comprises six code libraries that collectively provide a toolkit for authoring modular CHEMs, supplying them with data and using them to undertake reproducible analyses. The framework supports integrations with existing digital services for collaborative software development and data archiving. We are currently applying the software framework to develop and apply utility mapping models in youth mental health. We assess the first set of CHEMs that we have developed with the framework for a utility mapping study as wholly meeting both transparency criteria (open access code and data and clear author contributions and beliefs), two reusability criteria (promotes generalisability and transferability and liberal terms of use) and one updatability criteria (infrastructure for model maintenance) and partially meeting the remaining updatability criterion (retesting and deprecation). The assessment criteria and the software framework we have developed can inform future work to understand and improve ethical computational implementations of health economic models.

Code: Visit https://www.ready4-dev.com for more information about how to find, install and apply the prototype software framework.

[1] School of Public Health and Preventive Medicine, Monash University, Clayton, Australia
[2] Orygen, Parkville, Australia
[3] Centre for Youth Mental Health; The University of Melbourne, Parkville, Australia
[4] Heart Foundation, Melbourne, Australia
[5] headspace National Youth Mental Health Foundation, Melbourne, Australia
[6] Victoria University, Footscray, Australia
[7] Flinders University, Adelaide, Australia

✉ Correspondence: Matthew P Hamilton <matthew.hamilton1@monash.edu>

1. Introduction

Health economics is a discipline concerned with problems that arise due to scarce resources, such as how to value health and healthcare, allocate healthcare budgets and configure health services [1]. In seeking to solve these problems, health economists typically use models which are simplified and selective representations of systems that are believed to influence human health. A health economic model should be capable of being described using words and figures (a conceptual model), equations (a mathematical model) and computer code (a computational model). Health economic scientific manuscripts typically describe a model using its conceptual and mathematical representations but report results that have been generated by its computational representation (i.e., through execution of a computer program that applies the computational model to selected input data). The conceptual, mathematical and computational representations of a model are assumed to be isomorphic. Independent assessment of the validity of this assumption depends in part on how a computational model has been authored and shared.

Computational health economic models (CHEMs) can be implemented using specialized commercial software or authored as software development projects in an open-source programming language such as Python [2] or R [3]. Advantages of commercial modelling tools include simplicity and ease of use, but an open-source software development approach may facilitate computational models that are more transparent, reusable and updatable [4,5]. The decision about what type of implementation to choose for a specific CHEM project needs to balance both desired model features and the resource and time constraints within which the project needs to be delivered (see Table **1**).

For open-source CHEM projects, a major consideration is the selection of an appropriate software framework. A software framework is a shared common technology used by developers to collaboratively author software and which is not typically visible to software end-users [6]. A software framework provides a foundation for developing multiple software applications with shared resources, e.g., code and data files, that can be modified to suit specific needs. Software frameworks have widely been developed and implemented in data science, for example, PyTorch [7] for machine learning using Python.

Advantages of using software frameworks include facilitating code reuse and extension, promoting good programming practice and the capability to provide enhanced functionality and performance without additional effort by developers [8]. However, software frameworks can be challenging and time consuming to create [8] and then difficult for others to learn, often requiring model developers to undergo specialist training [6]. There is also a risk that a software framework may become excessively complex over time [8].

The ethical responsibilities of computational modellers are often poorly understood and inadequately fulfilled [9–12]. Guidance on developing, selecting and using software frameworks for ethical CHEM implementations is scarce. A high-level software framework for implementing open-source CHEMs in R has previously been developed with the primary aim of

improving model transparency [13]. However, open-source CHEMs remain rare [14,15], and novel software frameworks may be necessary to help health economists fulfill a broader set of ethical responsibilities. Ideally, software frameworks for ethical CHEMs would provide tools that address known barriers to ethical computational modelling practice. The main barriers to open-source CHEMs identified in a 2020 survey of health economists concern updating models, legal considerations, transferring model data, the platforms or languages in which models are developed, the level of public access required, lack of interest from decision-makers, model generalisability and confidentiality and security considerations [5].

In this paper, we describe:

(i) criteria for assessing ethical CHEM implementations;

(ii) a prototype software framework for the ethical implementation of CHEMs; and

(iii) use of the software framework to develop an open-source CHEM in youth mental health, with an initial focus on outcome valuation.

2. Assessing ethical CHEM implementations

We considered prior literature on modelling practice and our own professional experience to identify: (i) some core ethical responsibilities of CHEM developers; (ii) attributes of CHEMs that can suggest fulfillment of these responsibilities; and (iii) criteria against which these attributes can be assessed.

2.1. Ethical responsibilities of CHEM developers

We suggest that three core ethical responsibilities of health economic modellers are to:

- **make socially acceptable value judgments** (when planning CHEM projects);
- **ensure model fitness for purpose** (when implementing CHEMs); and
- **promote socially beneficial use of models** (once CHEMs have been released).

There is significant scope for health economists to improve fulfilment of each of these responsibilities.

Misalignment between the values of computational model developers and those of the population groups affected by decisions based on their models presents significant ethical risks [11,16]. The value judgments of CHEM developers influence the assumptions, selection of model features and standards for evidence that shape the model development process [17]. These value judgments are rarely made explicit, omissions that may lead to socially unacceptable policy recommendations [16]. Ideally, it should be straightforward for CHEMs to be modified to reflect alternative value judgments. However, such adaptability depends in part on the approach taken to computational model implementation (see Table **1**).

Computational modellers have duties both to take sufficient care that a computational model is adequate for the explicit purpose for which it was developed and to provide potential third-party users with the means of assessing its adequacy for their proposed purposes [9,14,18,19]. However, it is common for health economic models to have serious methodological flaws [20,21]; insufficient validation [22–24], poor reproducibility [25–27]; and undeclared errors [28]. Appropriate computational implementation choices can help address many of these shortcomings, for example by automating quality assurance checks and facilitating manual reviews by third parties.

A health economic model will have limited public benefit if it not much used, if it is mis-used or when its acceptability and adequacy rapidly decay. Reuse of CHEMs as components of other models can potentially make model development more efficient [29,30]. However, health economic models face challenges related to transferability across jurisdictions [30] that create barriers to reuse. Without ongoing maintenance, a CHEM risks becoming less reliable with time [30] and is at risk of being deployed for purposes for which it is poorly suited [12]. Currently, health economic models are rarely implemented computationally in a manner that facilitates routine updates [31], thus limiting the temporal window within which a CHEM can be validly applied.

2.2. CHEM attributes associated with ethical modelling practice

Health economist responsibilities during the planning, implementation and use of CHEMs are easier to state than to measure. However, aspects of ethical modelling practice may be inferred from measurable attributes of CHEMs. As described in Table **2**, we believe that the creators of transparent, reusable and updatable (TRU) CHEMs are likely to have fulfilled a number of their ethical obligations. We therefore selected these model attributes to use as the basis for deriving ethical assessment criteria.

Taking steps to make computational models **transparent** is an important part of ethical public health modelling practice [9]. Guidance on transparency in health economic modelling recommended that model code and data should be clearly documented [32]. Notably, the same guidelines, published over ten years ago, did not recommend sharing model code and data. However, more recent healthcare modelling guidance does recommend publicly dissemination of such artefacts [18]. Online repository services such as Zenodo [33] and Dataverse [34] provide persistent storage solutions that generate a Digital

Object Identifier (DOI) for each code and data collection. Ensuring that calculations are correct and consistent with model specifications is an essential part of CHEM quality assurance [35]. The extensiveness of such verification checks can be reported using the concept of code coverage [36] - the proportion of model code that has been explicitly tested. Tests should ideally combine both unit tests (to verify that small, isolated sections of code produce the correct output when run independently) and acceptance tests (to verify that the correct output is produced when multiple code components are run together to perform tasks that meet core user-requirements [37]). The nature and extent of individual model authorship contributions can become unclear when models are implemented over longer time-frames with a large and changing group of collaborators [11]. This issue can be addressed by use of online repository services such as GitHub [38], that provide citation tools and can transparently record all individual code contributions to a modelling project over its lifecycle.

A CHEM that is **reusable** also signals ethical modelling practice. Making a CHEM's code, data and documentation publicly available is increasingly considered good practice, yet is not sufficient to promote model re-use. Key concepts in health economic model reuse include generalisability (application without adaptation) and transferability (selective reuse and/or modification of model components) [39]. Writing model algorithms as collections of functions (short, self-contained and reusable software routines that each perform a discrete task) is good scientific computing practice [40] and promotes selective reuse. Computational implementations that store model code and data in distinct files and locations (as opposed to embedding data such as parameter values into source code) are easier to selectively modify. Modular implementations construct models from multiple independently reusable and replaceable sub-models (modules) [41]. The programming concept inheritance [42], deployed in object-oriented programming approaches, can facilitate the duplication and selective modification of models. Granting permissions to others to use, test and adapt models and their components, can be facilitated by two broad categories of open source licenses. Some guidance strongly recommends the use of permissive licensing [40] that provides users with great flexibility as to the purposes (including commercial) for which content can be re-used. An alternative approach is to use copyleft licenses [43] that can require content users to distribute any derivative works they create under similar open-source arrangements.

Models should be **updatable** so that they remain valid for longer, evolving as new evidence emerges and the systems being modelled change [30,44]. Ensuring that a model is regularly reviewed to identify and implement required improvements is a recommended defense against model validity decay [12]. Sustainable maintenance of open source research software requires both a core development team and an active user community [45]. Online communities can be an efficient means of engaging model users in testing each version of a model, identifying issues and suggesting improvements. Services such as GitHub [38] provide collaborative code development tools [46] that help elicit, integrate and reconcile contributions from multiple contributors and to ensure each update is uniquely identifiable and retrievable. It is also important that verification checks are rerun with each model update, a task that can be automated using the software development practice of continuous integration

[47]. The risk of model revisions having unintended consequences for third party users can be mitigated through the use of deprecation conventions [48] that take an informative and staged approach to retiring outdated model code and data.

2.3. Assessment criteria

For each CHEM attribute that suggests ethical CHEM modelling practice, we identified assessment criteria that have the potential to be met with the assistance of software frameworks.

How **transparent** a CHEM is can be assessed against the criteria:

- **T1**: All model code, non-confidential data and testing procedures and outcomes are available in open access repositories.
- **T2**: It is easy to see who developed and tested each part of the CHEM and to identify the modelling team's assumptions, judgments and theories about CHEM development and use.

How **reusable** a CHEM is can be assessed against the criteria:

- **R1**: Model code and data are implemented to facilitate both generalisability and transferability.
- **R2**: Terms of use allow anyone to reuse model code and non-confidential data, in whole or in part, without charge, and for purposes that include the creation of derivative works.

How **updatable** a CHEM is can be assessed against the criteria:

- **U1**: Technical infrastructure is in place to support version control and collaboration with model users in the maintenance of a model.
- **U2**: Each new release of a model is retested, with changes implemented to minimize disruptions for existing model users.

3. Software framework

To support the development of CHEMs that meet TRU assessment criteria, we have created a prototype software framework called ready4. The software framework aims to provide a toolkit for: (i) enabling modular implementation of CHEMs; (ii)

authoring and documenting CHEM modules; (iii) managing the labelling and transfer of CHEM input and output data; and (iv) authoring reproducible analyses that apply CHEMs to compatible data.

To achieve these goals, we have implemented the software framework as R [3] code libraries that integrate with a number of online services and which are supported by a documentation website.

3.1. R libraries

A library in the R language will typically depend on multiple other R libraries, all of which potentially having different authors. As the number of third-party dependencies of an R library grows so does the fragility of that library (e.g. the library may cease to work as intended due to changes in one of its dependency libraries). To reduce the fragility of our framework we implemented it as multiple R libraries rather than one R library. In total we authored six novel R libraries to implement the ready4 framework, all of which have distinct purposes and dependencies (Table **3**).

One framework library provides a **foundation for modular CHEM implementations**. In modular model implementations, modules need to be able to share inputs and outputs with each other and to be run as independent models [49]. To achieve this, the foundation framework library defines a template CHEM module (using R's S4 class system), which can be used to create other CHEM modules with a common set of inherited properties. One of these inherited properties is a novel syntax of 15 core commands that enable CHEM module algorithms to be consistently named. The foundation library also contains tools for retrieving web-based information on CHEM modules, datasets and analysis programs authored with the framework and for partially automating updates to a project documentation website.

Three framework libraries are designed to help streamline and standardise workflows for **authoring CHEM modules** from the template module. The R language supports functional and object-oriented programming paradigms [50]. Authoring with each of these paradigms is facilitated by a dedicated CHEM module authoring library. One module authoring library contains tools for writing functions in a consistent house style and then using the standardised naming conventions of that house style to automatically generate basic documentation for each function. A second module authoring library contains tools to help streamline and standardise the authoring and documenting of novel CHEM modules. A third module authoring library provides tools for disseminating themed bundles of CHEM modules as R libraries that are:

- documented (with a website and PDF manuals);
- licensed (using the copyleft GNU GPL-3 [66] by default);
- easily citable (citation information can be retrieved within an R session or from hosting repositories); and

- quality assured (each update triggers continuous integration workflows, including any acceptance and unit tests created by module library authors).

A library for **managing CHEM data** contains tools for supplying CHEM modules with input data ingested from local (i.e. a user's computer) or remote (online repositories) locations, labelling CHEM module datasets and exporting CHEM module data to online repositories. A library for **authoring reproducible analyses** contains tools to help write programs that apply CHEM modules to compatible datasets for the purpose of undertaking health economic analyses. These analysis programs can be either self-documenting (code is integrated with plain English explanations of what it does) or trigger the creation of separate documents (e.g. a scientific manuscript).

3.2. Online services

Framework libraries are designed to be used in conjunction with a number of third-party online services that we established and configured accounts with.

We created a GitHub organisation (a collection of code repositories) where code (libraries, programs and sub-routines) that we author for and with the framework is stored and version controlled. We configured individual repositories in our GitHub organisation to use GitHub actions to implement continuous integration. By default, code libraries authored with our framework will use continuous integration to assess compliance with policies specified by the Comprehensive R Archive Network (CRAN) [51]. To track our code coverage, we linked our GitHub organisation to an account we established at codecov [52]. To facilitate the creation and hosting of documentation websites, we enabled GitHub Pages in each repository used for code library development.

We also created a Zenodo community - a collection of permanent, uniquely identified repositories. We then linked our Zenodo community and GitHub organisation so that every time we specify a version of code in one of our GitHub repositories as a "release", a copy of that code is automatically created on Zenodo with a DOI. Finally, to manage model datasets, we created a dedicated collection within the Harvard Dataverse installation.

3.3. Documentation website

We developed a framework documentation website (www.ready4-dev.com) that provides guidance to model developers on how to use and contribute improvements to the ready4 software framework and models developed with it. The documentation website is versioned, which means documentation relating to prior versions of framework software can be archived, retrieved and viewed.

The documentation website was developed using the Hugo framework [53], Docsy theme [54] and Algolia DocSearch [55] and is hosted using the Netlify [56] service. We used functions from our foundation framework library to partially automate website updates relating to available CHEM modules, datasets and analysis programs. We linked our Netlify account to our GitHub organisation so that the project website would automatically update whenever its source code (publicly available in a GitHub repository) was edited.

4. Application

We plan on using the ready4 software framework to implement multiple transparent, reusable and updatable CHEMs in youth mental health (see Figure **1**). Some initial outputs from this work are publicly available.

4.1. Economic topics

Currently, we are using the ready4 software framework to develop, apply and share youth mental health CHEMs in four of the twelve domains of health economics identified by Wagstaff and Culyer [1]:

- health and its value (our projects: utility mapping models);
- determinants of health and ill-health (our projects: models for creating synthetic household populations with key risk and protective factors for mental disorders);
- demand for health and health care (our projects: spatial epidemiology and help-seeking choice models); and
- supply of health services (our projects: a model of primary mental health care services for young people).

Potential future directions are to supplement this work with CHEMs in two additional Wagstaff and Culyer domains of public health (to model the impact of selected fiscal policy and regulation options on young people's mental health) and human resources (to model the supply and behaviours of the youth mental health workforce). Our ultimate aim is to flexibly combining all our CHEMs in analyses that help answer questions in two additional Wagstaff and Culyer domains:

- efficiency and equity (our goal: assess the distributional impacts and identify the optimal targeting of youth mental health interventions); and
- economic evaluation (our goal: assess the cost-utility of competing policy options for improving the mental health of young people).

Although principally interested in using our CHEMs to answer policy questions relating to the mental health of young people in Australia, we want to facilitate CHEM transferability to other jurisdictions. Our CHEMs are therefore being derived from and applied to real data (which can be assumption, empirical or simulated, so long as it is appropriate for use in analysis intended to inform decision-making) from Australia. Additionally, we plan for all our CHEMs to be distributed with toy (fake) data to help demonstrate the potential use of these CHEMS in other decision contexts.

4.2. Case study: health and its value

We have previously described [57] a study to develop utility mapping models for use in samples of young people presenting to primary mental health services. The ready4 software framework was used in that study to develop CHEM modules, supply those modules with data and implement modelling analyses, creating the following artefacts:

- development version module libraries for describing and validating youth mental health human record datasets [58], scoring health utility [59], specifying utility mapping models [60] and implementing reproducible utility mapping studies [61];
- a development version library of functions for finding and using utility mapping models developed with these tools [62];
- collections of real data (study input and results [63]) and fake data (synthetic populations for testing model modules [64]);
- programs for replicating all steps from data ingest to manuscript reporting [65], applying utility mapping models to new data [66] and generating a synthetic representation of the study dataset [67]; and
- subroutines for creating a catalogue of utility mapping models [68] and generating a draft scientific manuscript [69] for studies implemented with these modules.

We created a checklist (Table **4**) that we used to subjectively assess these study outputs against TRU criteria. For each criterion, we provided a global assessment of whether it was met using the responses “yes”, “no” or “partial”. We believe the outputs from our utility mapping study may be assessable as having satisfactorily met five of the six criteria (T1, T2, R1, R2 and U1) and to have partially met one criterion (U2). The main shortcomings that we identified when applying the assessment criteria was that we have yet to adequately implement unit testing of the R libraries we authored as part of this study.

5. Discussion

Ethical practice is a core expectation of health researchers and computational methods underpin most quantitative research, yet an understanding of what constitutes ethical computational modelling practice in health is underdeveloped [9]. The modeller responsibilities, enabling model attributes and model implementation assessment criteria that we propose can help address this gap.

The ethical responsibilities and enabling model attributes we describe have both commonalities and distinctive features compared to a previous ethical framework for computational modelling in public health [9]. The authors of that framework propose 13 questions to evaluate ethical risk across the four criteria of independence, transparency, beneficence and justice. Their descriptions of the four criteria at least partially map to either our proposed modeler responsibilities ("justice" to "social acceptability", "independence" to "adequacy for purpose" and "beneficence" to "beneficial impact") or enabling model attributes ("transparency"). However, while our six assessment criteria are specific to three attributes (TRU) of the computational implementation of the model, the prior ethical framework includes questions relevant to the conceptual and mathematical models and the potential impacts of model use. Examples of these more general evaluation questions include (for the justice criterion) "is any lack of knowledge about important parameters attributable to uncertainty or variability?" and (for the beneficence criterion) "if a policy is based on the model evidence, is it more likely to be effective and beneficial than a decision made in the absence of the model?". The less numerous and more focused assessment criteria we propose may potentially be more tractable to implement in reviews of models authored by third parties and as the basis for designing software frameworks to support ethical computational model implementation.

Currently, many if not most existing CHEMs are insufficiently transparent [22,25–27], reusable [14,15] and updatable [31,70]]. Existing incentive structures for health economists generally do not promote facilitating peers to reuse their work. Currently, it can take "an extraordinary amount of idealism" to commit to authoring and maintaining research software [71].

Reducing waste in research is a core responsibility of research funders [72] and funding the development of CHEMs that are not adequately understood, reused or updated is wasteful. Previously recommended strategies for more beneficial health economic research investments include support for harmonized ethical standards for model development [9], methodological innovation to improve model transferability [73], networks of modellers working on common health conditions [74], and centralized infrastructure such as open source model repositories [5] and a standard platform for model implementations [22]. Development of software frameworks to support ethical CHEM implementations could enable and enhance each of these strategies.

As illustrated by Table **4**, we have developed a software framework that can help us to author a youth mental health model that largely satisfy our TRU criteria. However, we believe our software framework is currently too fragile to be anything more than a prototype for supporting the development needs other modelling teams and projects. A major reason for this distinction is that our software framework was developed with the needs of only one group of developers in mind – ourselves. We currently lack the resources required to adequately implement strategies to target factors such as user enjoyment, usability, active user-community and supporting resources that influence adoption of software frameworks [6].

Our prototype framework has a number of features that subsequent work to develop ethical software frameworks may find useful to incorporate. Firstly, developing a software framework to work within an existing and widely used open source programming language such as R or python, can keep framework scope relatively narrow (making it more tractable to develop, maintain and learn) while readily leveraging and coherently integrate other modelling tools written in that language (e.g. the dependency libraries we list in Table **3**). Secondly, implementation that combines both object oriented and functional programming paradigms can avail of the modular and syntactical simplicity benefits of the former, while limiting needless bundling of code artefacts. Thirdly, a sensible trade-off needs to be found between transparent code implementation (which requires clear and sufficiently detailed documentation) and Agile Software Development (for which a foundational principle is prioritizing the development of working code over writing documentation [75]). Our software framework makes this trade off by enforcing the use of consistent code naming conventions and file organisation which in turn enables automated generation of simple documentation at every code update. All model data-structures and algorithms are therefore always documented (at least minimally, with machine authored content), meaning model developers have a requirement to write customized documentation less frequently.

A future software framework for ethical CHEMs would ideally incorporate a base set of features useful to developers of computational models across all domains of public health, with the capability for community-led extensions that are tailored to the needs of modellers focused on specific health-conditions.

6. Conclusion

We have identified criteria that can be used to systematically assess extent to which the computational implementation of health economic models adheres to the ethical goals of transparency, reusability and updatability. We have developed an open-source software framework that can support the ethical computational implementation of economic models in youth mental health. Our framework can be used as a prototype for developing future software frameworks to support ethical implementation of CHEMs.

Acknowledgement

The authors would like to acknowledge the contribution of John Gillam who provided advisory input to this research.

Availability of data and materials

The most up to date and comprehensive source of documentation on our framework and model is available at https://www.ready4-dev.com . Development versions of all code repositories referenced in this article are available in https://github.com/ready4-dev/ . Archived code releases are available in https://zenodo.org/communities/ready4 . All data repositories referenced in this article are available in https://dataverse.harvard.edu/dataverse/ready4 .

Ethics approval

Software framework development did not involve human subject research and was not ethically reviewed. The utility mapping worked example is a previously reported study that was reviewed and granted approval by the University of Melbourne's Human Research Ethics Committee, and the local Human Ethics and Advisory Group (1645367.1).

Funding

Software framework development was funded by Orygen, VicHealth, Victoria University and an Australian Government Research Training Program (RTP) Scholarship . The utility mapping study used as a worked example was funded by the National Health and Medical Research Council (NHMRC, APP1076940), Orygen and headspace.

Conflict of Interest

None declared.

References

1. Wagstaff A, Culyer AJ. Four decades of health economics through a bibliometric lens. Journal of health economics. Elsevier; 2012;31: 406–439.

2. Van Rossum G, Drake FL. Python 3 reference manual. Scotts Valley, CA: CreateSpace; 2009.

3. R Core Team. R: A language and environment for statistical computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2022. Available: https://www.R-project.org/

4. Incerti D, Thom H, Baio G, Jansen JP. R you still using excel? The advantages of modern software tools for health technology assessment. Value in Health. Elsevier; 2019;22: 575–579.

5. Pouwels X, Sampson CJ, Arnold RJG. Opportunities and barriers to the development and use of open source health economic models: A survey. Value Health. 2022;25: 473–479. doi:10.1016/j.jval.2021.10.001

6. Myllärniemi V, Kujala S, Raatikainen M, Sevońn P. Development as a journey: Factors supporting the adoption and use of software frameworks. Journal of software engineering research and development. SpringerOpen; 2018;6: 1–22.

7. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. Proceedings of the 33rd international conference on neural information processing systems. Red Hook, NY, USA: Curran Associates Inc.; 2019.

8. Edwin NM. Software frameworks, architectural and design patterns. Journal of Software Engineering and Applications. Scientific Research Publishing; 2014;2014.

9. Boden LA, McKendrick IJ. Model-based policymaking: A framework to promote ethical “good practice” in mathematical modeling for public health policymaking. Frontiers in Public Health. 2017;5. doi:10.3389/fpubh.2017.00068

10. Pliakos EE. Ethics in economic modeling in health care. AMA Journal of Ethics. American Medical Association; 2021;23: 599–600.

11. Thompson E. Escape from model land: How mathematical models can lead us astray and what we can do about it. New Yourk: Basic Books; 2022.

12. Calder M, Craig C, Culley D, De Cani R, Donnelly CA, Douglas R, et al. Computational modelling for decision-making: Where, why, what, who and how. Royal Society open science. The Royal Society Publishing; 2018;5: 172096.

13. Alarid-Escudero F, Krijkamp EM, Pechlivanoglou P, Jalal H, Kao S-YZ, Yang A, et al. A need for change! A coding framework for improving transparency in decision modeling. Pharmacoeconomics. Springer; 2019;37: 1329–1339.

14. Feenstra T, Corro-Ramos I, Hamerlijnck D, Voorn G van, Ghabri S. Four aspects affecting health economic decision models and their validation. PharmacoEconomics. 2022;40: 241–248. doi:10.1007/s40273-021-01110-w

15. Emerson J, Bacon R, Kent A, Neumann PJ, Cohen JT. Publication of decision model source code: Attitudes of health economics authors. PharmacoEconomics. 2019;37: 1409–1410. doi:10.1007/s40273-019-00796-3

16. Duckett S. A journey towards a theology of health economics and healthcare funding. Theology. SAGE Publications Sage UK: London, England; 2022;125: 326–334.

17. Harvard S, Werker GR, Silva DS. Social, ethical, and other value judgments in health economics modelling. Social Science & Medicine. 2020;253: 112975. doi:https://doi.org/10.1016/j.socscimed.2020.112975

18. Erdemir A, Mulugeta L, Ku JP, Drach A, Horner M, Morrison TM, et al. Credible practice of modeling and simulation in healthcare: Ten rules from a multidisciplinary perspective. Journal of translational medicine. 2020;18: 369. doi:10.1186/s12967-020-02540-4

19. Thompson EL, Smith LA. Escape from model-land. Economics. De Gruyter Open Access; 2019;13.

20. Carletto A, Zanuzzi M, Sammarco A, Russo P. Quality of health economic evaluations submitted to the italian medicines agency: Current state and future actions. International Journal of Technology Assessment in Health Care. Cambridge University Press; 2020;36: 560–568. doi:10.1017/S0266462320000641

21. Wonder M, Dunlop S. Assessment of the quality of the clinical evidence in submissions to the australian pharmaceutical benefits advisory committee: Fit for purpose? Value in Health. 2015;18: 467–476. doi:https://doi.org/10.1016/j.jval.2015.02.011

22. Ghabri S, Stevenson M, Möller J, Caro JJ. Trusting the results of model-based economic analyses: Is there a pragmatic validation solution? Pharmacoeconomics. 2019;37: 1–6. doi:10.1007/s40273-018-0711-9

23. Kolovos S, Bosmans JE, Riper H, Chevreul K, Coupé VM, Tulder MW van. Model-based economic evaluation of treatments for depression: A systematic literature review. PharmacoEconomics-open. Springer; 2017;1: 149–165.

24. Haji Ali Afzali H, Gray J, Karnon J. Model performance evaluation (validation and calibration) in model-based studies of therapeutic interventions for cardiovascular diseases: A review and suggested reporting framework. Applied health economics and health policy. Springer; 2013;11: 85–93.

25. Jalali MS, DiGennaro C, Guitar A, Lew K, Rahmandad H. Evolution and reproducibility of simulation modeling in epidemiology and health policy over half a century. Epidemiologic Reviews. 2021;43: 166–175. doi:10.1093/epirev/mxab006

26. McManus E, Turner D, Sach T. Can you repeat that? Exploring the definition of a successful model replication in health economics. Pharmacoeconomics. 2019;37: 1371–1381. doi:10.1007/s40273-019-00836-y

27. Bermejo I, Tappenden P, Youn J-H. Replicating health economic models: Firm foundations or a house of cards? PharmacoEconomics. 2017;35: 1113–1121. doi:10.1007/s40273-017-0553-x

28. Radeva D, Hopkin G, Mossialos E, Borrill J, Osipenko L, Naci H. Assessment of technical errors and validation processes in economic models submitted by the company for NICE technology appraisals. International Journal of Technology Assessment in Health Care. 2020;36: 311–316. doi:10.1017/S0266462320000422

29. Arnold RJG, Ekins S. Time for cooperation in health economics among the modelling community. PharmacoEconomics. 2010;28: 609–613. doi:10.2165/11537580-000000000-00000

30. Garcia-Mochon L, Rovira Forns J, Espin J. Cost transferability problems in economic evaluation as a framework for an european health care and social costs database. Cost Effectiveness and Resource Allocation. Springer; 2021;19: 1–12.

31. Sampson CJ, Wrightson T. Model registration: A call to action. PharmacoEconomics - Open. 2017;1: 73–77. doi:10.1007/s41669-017-0019-2

32. Eddy DM, Hollingworth W, Caro JJ, Tsevat J, McDonald KM, Wong JB. Model transparency and validation: A report of the ISPOR-SMDM modeling good research practices task force-7. Med Decis Making. 2012;32: 733–43. doi:10.1177/0272989x12454579

33. European Organization For Nuclear Research, OpenAIRE. Zenodo [Internet]. CERN; 2013. doi:10.25495/7GXK-RD71

34. Quantitative Social Science I for. Dataverse [Internet]. Harvard University; 2007. Available: https://dataverse.org

35. Büyükkaramikli NC, Rutten-van Mölken MPMH, Severens JL, Al M. TECH-VER: A verification checklist to reduce errors in models and improve their credibility. PharmacoEconomics. 2019;37: 1391–1408. doi:10.1007/s40273-019-00844-y

36. Eric Wong W, Debroy V, Choi B. A family of code coverage-based heuristics for effective fault localization. Journal of Systems and Software. 2010;83: 188–208. doi:https://doi.org/10.1016/j.jss.2009.09.037

37. Martin RC. Agile software development: Principles, patterns, and practices. Prentice Hall PTR; 2003.

38. github. GitHub [Internet]. 2007. Available: https://github.com/

39. Augustovski F, Iglesias C, Manca A, Drummond M, Rubinstein A, Marti SG. Barriers to generalizability of health economic evaluations in latin america and the caribbean region. Pharmacoeconomics. 2009;27: 919–29. doi:10.2165/11313670-000000000-00000

40. Wilson JAC Greg AND Bryan. Good enough practices in scientific computing. PLOS Computational Biology. Public Library of Science; 2017;13: 1–20. doi:10.1371/journal.pcbi.1005510

41. Pan M, Gawthrop PJ, Cursons J, Crampin EJ. Modular assembly of dynamic models in systems biology. PLoS computational biology. Public Library of Science San Francisco, CA USA; 2021;17: e1009513.

42. Hourani H, Wasmi H, Alrawashdeh T. A code complexity model of object oriented programming (OOP). 2019 IEEE jordan international joint conference on electrical engineering and information technology (JEEIT). 2019. pp. 560–564. doi:10.1109/JEEIT.2019.8717448

43. Foundation TFS. What is copyleft? [Internet]. Available: https://www.gnu.org/copyleft/

44. Jenkins DA, Martin GP, Sperrin M, Riley RD, Debray TPA, Collins GS, et al. Continual updating and monitoring of clinical prediction models: Time for dynamic prediction systems? Diagnostic and Prognostic Research. 2021;5: 1. doi:10.1186/s41512-020-00090-3

45. Ye Y, Barapatre S, Davis MK, Elliston KO, Davatzikos C, Fedorov A, et al. Open-source software sustainability models: Initial white paper from the informatics technology for cancer research sustainability and industry partnership working group. J Med Internet Res. 2021;23: e20028. doi:10.2196/20028

46. Mergel I. Open collaboration in the public sector: The case of social coding on GitHub. Government Information Quarterly. 2015;32: 464–472. doi:https://doi.org/10.1016/j.giq.2015.09.004

47. Shahin M, Ali Babar M, Zhu L. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. IEEE Access. 2017;5: 3909–3943. doi:10.1109/ACCESS.2017.2685629

48. Zhou J, Walker RJ. API deprecation: A retrospective analysis and detection method for code examples on the web. Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering. 2016. pp. 266–277.

49. Barros C, Luo Y, Chubaty AM, Eddy IM, Micheletti T, Boisvenue C, et al. Empowering ecological modellers with a PERFICT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation. Methods in Ecology and Evolution. Wiley Online Library; 2023;

50. Chambers JM. Object-Oriented Programming, Functional Programming and R. Statistical Science. Institute of Mathematical Statistics; 2014;29: 167–180. doi:10.1214/13-STS452

51. Statistical Computing RF for. The comprehensive r archive network [Internet]. 2022. Available: https://cran.r-project.org

52. Codecov [Internet]. Available: https://about.codecov.io/

53. Hugo: The world's fastest framework for building websites [Internet]. Available: https://gohugo.io

54. Docsy [Internet]. Available: https://www.docsy.dev

55. Docsearch [Internet]. Available: https://docsearch.algolia.com

56. Netlify [Internet]. Available: https://www.netlify.com

57. Hamilton MP, Gao CX, Filia KM, Menssink JM, Sharmin S, Telford N, et al. Mapping psychological distress, depression and anxiety measures to AQoL-6D utility using data from a sample of young people presenting to primary mental health services. medRxiv. Cold Spring Harbor Laboratory Press; 2022; doi:10.1101/2021.07.07.21260129

58. Hamilton M, Gao C. youthvars: Describe and Validate Youth Mental Health Datasets [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6084467

59. Hamilton M, Gao C. Scorz: Score questionnaire item responses [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6084824

60. Hamilton M, Gao C. specific: Specify Candidate Models for Representing Mental Health Systems [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6116701

61. Gao C, Hamilton M. TTU: Implement Transfer to Utility Mapping Algorithms [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6130155

62. Hamilton MP, Gao CX. Youthu: Transform youth outcomes to health utility predictions [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6210978

63. Hamilton MP, Gao CX, Filia KM, Menssink JM, Sharmin S, Telford N, et al. Transfer to AQoL-6D Utility Mapping Algorithms [Internet]. Harvard Dataverse; 2021. doi:10.7910/DVN/DKDIB0

64. Hamilton MP. Synthetic (fake) youth mental health datasets and data dictionaries [Internet]. Harvard Dataverse; 2021. doi:10.7910/DVN/HJXYKQ

65. Hamilton M, Gao C. Complete study program to reproduce all steps from data ingest through to results dissemination for a study to map mental health measures to AQoL-6D health utility [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6212704

66. Hamilton M, Gao C. aqol6dmap_use: Apply AQoL-6D Utility Mapping Models To New Data [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6416330

67. Hamilton MP. aqol6dmap_fakes: Generate fake input data for an AQoL-6D mapping study [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6321821

68. Hamilton M. ttu_mdl_ctlg: Generate a template utility mapping (transfer to utility) model catalogue [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6116385

69. Hamilton MP. ready4-dev/ttu_lng_ss: Create a Draft Scientific Manuscript For A Utility Mapping Study [Internet]. Zenodo; 2022. doi:10.5281/zenodo.5976988

70. Kokol P, Blažun Vošner H, Kokol M, Završnik J. Role of agile in digital public health transformation. Frontiers in Public Health. 2022;10. doi:10.3389/fpubh.2022.899874

71. Anzt H, Bach F, Druskat S, Löffler F, Loewe A, Renard BY, et al. An environment for sustainable research software in germany and beyond: Current state, open challenges, and call for action. F1000Research. Faculty of 1000 Ltd; 2020;9.

72. Chalmers I, Bracken MB, Djulbegovic B, Garattini S, Grant J, Gülmezoglu AM, et al. How to increase value and reduce waste when research priorities are set. The Lancet. Elsevier; 2014;383: 156–165.

73. Craig P, Di Ruggiero E, Frolich KL, Mykhalovskiy E, White M, Campbell R, et al. Taking account of context in population health intervention research: Guidance for producers, users and funders of research. National Institute for Health Research; 2018;

74. Sampson CJ, Arnold R, Bryan S, Clarke P, Ekins S, Hatswell A, et al. Transparency in decision modelling: What, why, who and how? PharmacoEconomics. 2019;37: 1355–1369. doi:10.1007/s40273-019-00819-z

75. Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, et al. Manifesto for agile software development. Snowbird, UT; 2001;

Tables and figures

Table 1: *Considerations when choosing between implementing computational models using commercial modelling software or as software projects using an open-source programming language*

	Commercial modelling software	Open source programming language
Project costs	Default option for projects with tight time and resource constraints. Requires less extensive range of (potentially hard to find) specialized skills.	Initial development is likely to be time and resource intensive, though over medium to long term time horizons there is the potential for efficiency savings if: (i) model development can leverage artefacts (code and data) from pre-existing models; and / or (ii) project requirements include model maintenance and transfer.
Project quality	Robustness of commercial software means users can have high degrees of confidence that model files will open and execute correctly over medium to long-term. Lack of integration with tools used for scientific manuscript authoring may result in transcription errors when reporting results.	Facilitates more extensive / complex model representations and use in interactive decision aids. Supports model transparency, reusability and updatability. Lots of novel code is a potential source of errors in model implementations. Potentially fragile – if not maintained or bundled with all required and correctly versioned dependencies, models may not execute as intended in the future.

Table 2: *How transparent, reusable and updatable (TRU) computational health economic models (CHEMs) suggest ethical modelling practice*

<table>
<tr><th></th><th>Social acceptability</th><th>Fitness for purpose</th><th>Beneficial use</th></tr>
<tr><td>Transparent CHEMs</td><td colspan="2">Value judgments and assumptions can be reviewed by third parties.</td><td>Clarity about model features and performance reduces risk of inappropriate use in decision-making.</td></tr>
<tr><td>Reusable CHEMs</td><td rowspan="2">Use and modification of models by third-parties allows alternative value judgments to be explored.</td><td>Use by third parties increases likelihood of uncovering errors.</td><td>More potential model beneficiaries and less duplicative modeller effort.</td></tr>
<tr><td>Updatable CHEMs</td><td colspan="2">Well maintained models can be validly used for longer.</td></tr>
</table>

Table 3: *Software framework R libraries*

Library	Purpose	Dependency R libraries
ready4	Provide a template and novel syntax for modular CHEM implementations and tools for finding interoperable CHEM modules, datasets and reproducible analysis programs.	assertthat bib2df dataverse dplyr fs Hmisc kableExtra knitr lifecycle magrittr methods natmanager piggyback purrr readr readxl rlang rmarkdown rvest stats stringi stringr testit testthat tibble tidyRSS tools utils zen4R
READY4FUN	Streamline and standardise the authoring and documenting of functions that support transferable and generalisable model algorithms.	desc devtools dplyr generics gert Hmisc knitr lifecycle lubridate magrittr methods piggyback pkgdown purrr readxl ready4 ready4show ready4use rlang sinew stats stringi stringr testit testthat tibble tidyr tools usethis utils xfun
ready4class	Streamline and standardise the authoring and documenting of new interoperable CHEM modules.	devtools dplyr fs gtools Hmisc knitr lifecycle magrittr methods purrr ready4 ready4fun ready4show rlang stats stringi stringr testit testthat tibble tidyr usethis utils
ready4pack	Help bundle and disseminate newly created CHEM modules as R libraries that are documented, licensed and quality assured.	dataverse dplyr knitr lifecycle magrittr methods purrr ready4 ready4class ready4fun rlang stringr testthat tibble tidyr utils
ready4use	Help manage the labelling and transfer of data between CHEM modules and local and remote data repositories.	data.table dataverse dplyr fs Hmisc knitr lifecycle magrittr methods piggyback purrr readxl ready4 ready4show rlang stats stringi stringr testit testthat tibble tidyr utils
ready4show	Facilitate the use of CHEM modules in programs that make the entire data ingest, analysis and reporting pipeline reproducible can be challenging and time consuming.	dataverse DescTools dplyr flextable grDevices here Hmisc kableExtra knitr knitrBootstrap lifecycle magrittr methods officer purrr ready4 rlang rmarkdown stringi stringr testthat tibble tidyr utils xtable

Table 4: *Transparent, reusable and updatable (TRU) assessment criteria applied to outcome valuation computational health economic model (CHEM)*

Criteria	Met?	Detail
T1 Open access	Yes	All source code and testing procedures are available in public GitHub repositories, with each code release persistently available in a Zenodo repository. Study dataset contains confidential patient records and was not published. Instead, a synthetic representation of the study dataset is persistently available in a repository in the Harvard Dataverse. Data files to support out of sample application of models are published at the same location.
T2 Author contributions & beliefs	Yes	All module libraries, programs, and datasets are distributed with citation information. GitHub repositories detail author code contributions over project development history are publicly visible. Model catalogues persistently available on the Harvard Dataverse describe the predictive performance of models under multiple usage regimes. Each code library is documented with worked examples of how to apply modules. Analysis and reporting programs are self-documenting. Sub-routines for generating reports are documented with README files.
R1 Generalisability & transferability	Yes	Model code is written using both functional and object-oriented paradigms. Code library websites include hypothetical examples of generalisability (applying study algorithm to estimate mapping models from new data with the same predictor and outcome variables) and transferability (adapting study algorithm to develop mapping models from datasets with different predictor variables and outcomes measured with a different utility instrument).
R2 Open-source licenses	Yes	All code is distributed using GPL-3 licenses. Datasets use amended version of template provided by Harvard Dataverse, allowing reuse of data subject to some ethical restrictions (e.g., use in efforts to reidentify study participants is prohibited).
U1 Version-control and maintenance	Yes	All code is version controlled using git and GitHub, with semantic versioning. Each code library has a specified maintainer and guidance for potential code contributors is available on the project website.
U2 Retesting & Deprecation	Partial	Continuous integration used for all code libraries, primarily for acceptance testing. Only limited use is made of unit testing. Retired library code is deprecated using tools from the lifecycle R library. Library documentation articles and datasets are also deprecated.

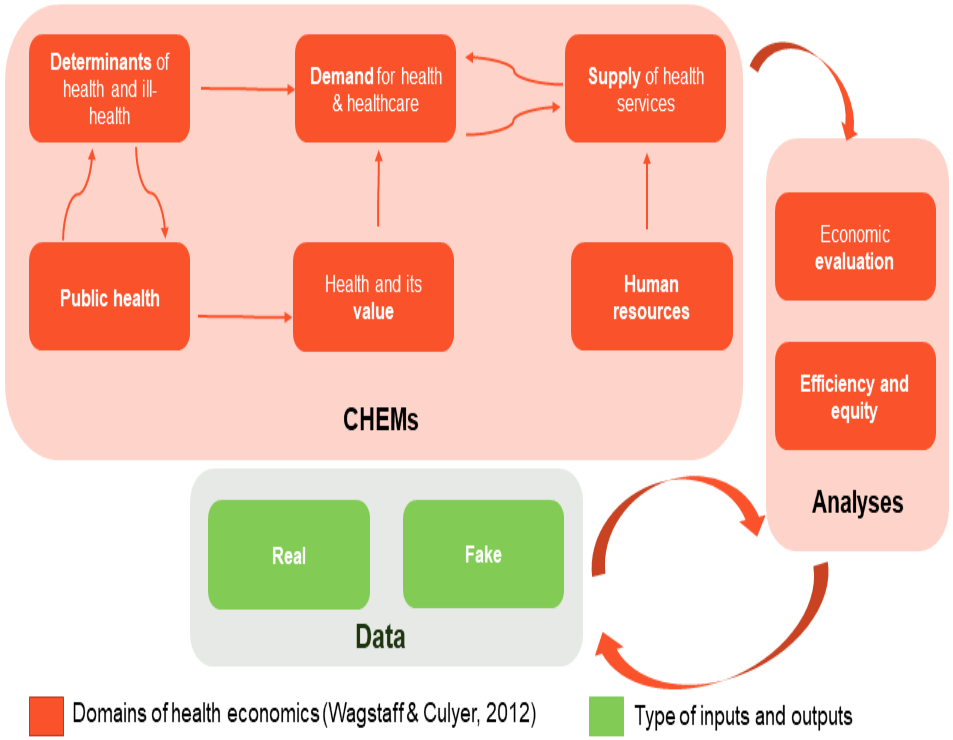

Figure 1: *High level summary of planned implementation of youth mental health economic model*