Financing digital health software global goods:

Summary of findings of relevant business models

December 2024

TABLE OF CONTENTS

BACKGROUND | Objectives

Digital health software global goods play a crucial role in digitalizing health systems—leading to improvements in access to, efficiency of, and quality of care, particularly in low-resource settings. However, the free and open-source status of global goods presents revenue challenges to the organizations* that develop and maintain them—threatening the long-term sustainability of global goods and their contributions to health systems.

The objectives of this analysis are to:

1. generate a **list of revenue, pricing, and cost models** relevant to global goods,
2. share **benefits, challenges, and considerations for global goods developers** for each model,
3. provide **key findings on challenges and opportunities** for global goods developers, and
4. recommend **next steps** for further analysis and validation.

*referred to as *developers*

BACKGROUND | Scope of business models considered

This study considered three types of business models:

- **Revenue models** outline the strategy to generate income from customers by offering products or services.
- **Pricing models** define how to set the price of a product or service, balancing factors like market demand, value delivered, and affordability.
- **Cost models** focus on reducing or managing expenses, leading to operational efficiencies and cost savings.

Revenue, pricing, and cost models may be used simultaneously or in combination.

BACKGROUND | Model definitions (1)

Revenue models	
Advertisement / sponsorship	Advertiser or sponsor pays to display advertisements within the digital tool.
Certification	Developer offers access to the digital tool for free, but customer pays to be certified in the use of the digital tool to demonstrate competency.
Core capability	Developer sells access to processes or systems developed to support the digital tool that can be replicated in other contexts (e.g., developer-owned cloud computing power is leased to customer for unrelated use case).
Customization	Customer pays developer to customize a standard digital tool.
Data monetization	Developer sells user data (which may be depersonalized) for advertising or other purposes.
Donor or grant funding	Developer receives term-limited funding to support the development, deployment, or ongoing maintenance of a digital tool.
Dual license	Customer can access the code for free, but the terms of use can change depending on how the software is used. This includes charging customers for the ability to modify, distribute, or monetize the digital tool.
Fundraise / crowdfund	Developer solicits donations or sells branded merchandise to support ongoing maintenance of the digital tool or to develop a new feature.
Gain sharing	Customer agrees to pay a portion of the monetary gains that come from implementation of the digital tool.

BACKGROUND | Model definitions (2)

Revenue models (continued)	
Microtask	User receives access to digital tool in exchange for completing a “microtask” that the developer can monetize (e.g., to access the tool, the user must identify objects in an image, with the output being sold to train machine learning models).
Software as a service (SaaS)	Customer pays fees to access cloud-based digital tool—usually through a subscription or pay-per-use.
Supporting service	Customer pays for complementary services to digital tool (e.g., training, data analysis, data visualization).
White label	Customer pays for unbranded digital tool—allowing another business to use it under their own brand name.
Pricing models	
Freemium	Developer offers a “basic” version of the digital tool for free, and a “premium” version for a fee.
Robin Hood	Developer charges a different price for the digital tool based on ability-to-pay (e.g., different price for customers in high-income countries vs. low-income countries).
Variable or unit-based	Customer pays for use of the digital tool based on number of units utilized during a specified duration (e.g., users per month).
Costing models	
Community support	A community of users supports the development, maintenance, and/or development of new features for the digital tool.

BACKGROUND | Key actors and customers

For this analysis, *customer* refers to any entity providing funding directly to the global good software developer.

Funders, government program owners, system implementers, and maintenance and support providers can all serve as customers*.

Customers may vary by model.

*The *advertisement / sponsorship* and *microtask* models may include additional customers beyond those listed here.

Actor	Role	Examples	Possible customer
Funder	Provides short or long-term funding	• Bilateral and multilateral aid agencies • Foundations • Governments	☑
Government program owner	Represents government program needs and oversees implementation	• Government ministries or agencies (national and/or subnational level)	☑
Software developer	Performs core software development and maintenance; may contribute to software customization and support	• Global software companies • Regional software companies • Local software companies • Government health information technology (IT) staff	
System implementer	Manages software adaptation against program goals and initial deployment, often including training, partner management, and support	• Global software companies • Regional software companies • Local software companies • Government health IT staff	☑
Maintenance and support provider	Provides ongoing system maintenance and support, including system and infrastructure upgrades	• Software / platform developer • System implementer • Government health IT staff	☑
System user	Uses the solution software; is often a healthcare actor	• Suppliers • Government system administrators • Central medical stores, laboratories • Warehouses, transport / logistics • Facility users (e.g., health workers)	

Adapted from: Vital Wave and Digital Square, 2020. *Development actor behavior through the lens of a value chain analysis*. Available at: https://digitalsquare.org/s/Digital-Health-Value-Chain-Assessment-Results.pdf.

METHODOLOGY

Literature review

Searched using Google Scholar:

- 2010 or later
- Terms: “software” AND “business model” OR “business models” OR “revenue model” OR “revenue models”

347 abstracts reviewed
20 full articles reviewed

Supplemented with information from select grey literature from Digital Square files.

Online survey

Survey respondents (n=**27**) reviewed a list of business models and corresponding definitions (slides 6-7) informed by the literature review.

For each model, respondents were asked to indicate if they had used or considered the model, and the benefits and challenges of the models they had used or considered.

Survey respondents included both global goods (n=23) and proprietary software (n=4) developers.

Virtual interviews

Select survey respondents (n=**16**) (including both global goods (n=15] and proprietary software (n=1] developers) were contacted for interviews to provide more detailed information about their responses.

Interviews were also conducted to address gaps in the understanding of business models.

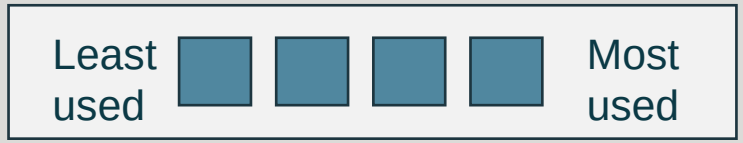
Analysis

Responses from the online survey and interview transcripts were collated by relevant model(s).

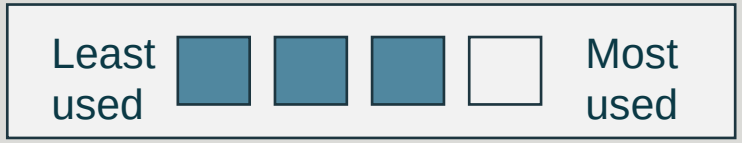
Responses were reviewed and coded as a benefit, challenge, or important consideration.

Other important information such as cost savings strategies, considerations for implementation, navigating the global goods ecosystem, and model innovations were also coded to inform key findings.

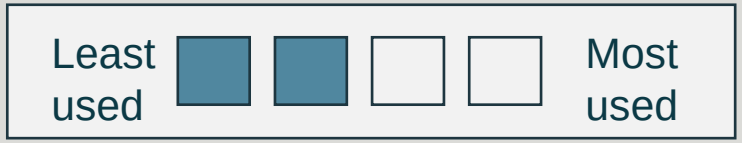
RESULTS | Frequency of use



customization, donor or grant funding, supporting service, community support



certification, fundraise / crowdfund, SaaS, robin hood



advertisement / sponsorship, dual license, variable or unit-based

core capability, data monetization, gain sharing, microtask, white label, freemium

RESULTS | Benefits, challenges, and considerations

Least used ■ ■ □ □ Most used

Advertisement / sponsorship

Definition: Advertiser or sponsor pays to display advertisements within digital tool.

Benefits	Challenges
• Requires minimal effort, as the developer simply displays the advertiser's generated content. • Can be paired with other models. • Provides income without charging users directly.	• Requires high user traffic to be enticing for advertisers. • Potentially reduces engagement with the digital tool, as it can be distracting to users. • Advertising companies may request user data for more targeted ads.

Considerations for global goods

- Developers may avoid working with advertisers requesting user data to protect patient privacy.
- Most respondents that had used this model were proprietary digital health tool developers.

Least used ■ ■ ■ □ Most used

Certification

Definition: Developer offers access to the digital tool for free, but customer pays to be certified in the use of the digital tool to demonstrate competency.

Benefits	Challenges
• Offers flexibility in how the model can be applied. Certification can be offered a range of ways: from a "certificate of completion" for introductory-level courses to becoming an "expert" user. • Ensures implementers or users meet certain standards when using a digital tool, which is beneficial to developers. • Benefits implementers or users as they can showcase their proficiency to funders.	• Requires support from the developer. Creating certification programs—either self-guided or instructor-led—can require developers to generate materials and have staff dedicated to certification programs. • Leads to risks for developers if "certified" users improperly use the digital tool.

Considerations for global goods

- Developers often use this model by offering "academy"-style services, which provide expert-led or self-paced courses on the digital tool, with certification awarded upon completion.
- Developers that offer training could consider creating a certification program—repurposing existing training materials.
- Respondents who had used the model indicated that they used certification as a supplementary revenue stream.

Least used ■ □ □ □ Most used

Core capability

Definition: Developer sells access to processes or systems developed to support the digital tool that can be replicated in other contexts (e.g., developer-owned cloud computing power is leased to customer for unrelated use case).

Benefits	Challenges
• Does not depend on delivering or providing a service for revenue generation. • Can supplement revenue generated from the digital tool itself.	• Potentially takes human resources away from the development, deployment, or maintenance of the digital tool.

Considerations for global goods

- Cloud computing is a potential core capability, but no developers reported selling this functionality to customers outside of those directly using their digital tool.
- Technical expertise (e.g., knowledge of logistics management) may also be considered core capability that is available. However, leveraging human resources for supplying services not directly related to the ongoing development and maintenance of the tool reduces staff capacity to work on the core digital tool.

Least used ■ ■ ■ ■ Most used

Customization

Definition: Customer pays developer to customize a standard digital tool.

Benefits	Challenges
• Leads to improvements when innovations from customizations are incorporated into the core digital tool. • Opens many opportunities for generating revenue as most customers do not have time or capacity to customize software to their needs. • Leads to increased user proficiency and customer loyalty by tailoring to user needs. • Benefits all customers as the core digital tool gains features. • Develops user expertise if they actively participate in the customization process—increasing local capacity.	• Does not always result in widely-applicable modifications to the core digital tool. • Takes resources away from core development as projects can be time consuming. • Requires more work than selling software “off the shelf” as digital tools are tailored to each customer. • Involves costs that are not always understood by funders who lack a realistic view of the effort required by the developer to customize a digital tool.

Customization (cont.)

Considerations for global goods

- Customizations are often a one-time charge, but there is potential for longer-term maintenance agreements or providing supporting services to customers who have paid for a customization.
- Developers should consider local implementers, as local implementers may have a stronger understanding of user needs. This ensures all contributors to the ecosystem have opportunities to sustain their organizations financially.
- Customization was one of the most frequently used revenue models amongst respondents.

Least used ■ □ □ □ Most used

Data monetization

Definition: Developer sells user data (which may be depersonalized) for advertising or other purposes.

Benefits	Challenges
• Can complement other revenue models.	• Requires depersonalization when relying on personally-identifiable data, which can be challenging to execute. • Is not permitted in many countries.

Considerations for global goods

- A data monetization model may be possible with data that lacks personally identifiable information, such as logistics data.
- Multiple developers said this was not a revenue model they considered for ethical reasons.
- No respondents had used this model.

Least used ■ ■ ■ ■ Most used

Donor or grant funding

Definition: Developer receives term-limited funding to support the development, deployment, or ongoing maintenance of a digital tool.

Benefits	Challenges
• Can boost a developer's credibility and reputation, attracting additional funding or partnerships from other sources. • Allows developers to focus on social impact projects. • Provides funding for digital tools that the market fails to support.	• Requires time intensive proposal writing, without the guarantee of an award. • Leaves developers vulnerable to shifts in donor priorities, which may fluctuate or differ from the developer's vision. • Is usually short term, especially funding is intended to *develop* digital tools.

Considerations for global goods

- Donors often support implementation of a digital tool, and do not typically provide funding or ongoing *development* or *maintenance* costs (i.e., operations costs).
- Donors provide nearly all funding for digital health software global goods.

Least used ■ ■ □ □ Most used

Dual license

Definition: Customers can access the code for free, but the terms of use can change depending on how the software is used. This includes charging customers for the ability to modify, distribute, or monetize the digital tool.

Note: A dual license model differs from the *Freemium* pricing model. A dual license model generates revenue when customers modify a tool whereas with a Freemium model monetizes a "premium" version of the tool built by the developer.

Benefits	Challenges
• Can attract a wider range of customers by offering a free version of the digital tool. Attracting a wider customer base also creates brand recognition. • Generates revenue from customers willing to pay for modification, distribution, or monetization of the digital tool.	• Requires determining which digital tool functionalities will be available for free, a process that can be complex. • Requires balancing the interest of free access customers and paying customers. • Could have negative consequences on user engagement if a tool was initially offered for free.

Considerations for global goods

- Digital tools can remain a global good while under a dual license if one license option remains free and open source. Open-source licenses allow for customers to use, modify, and share software for free under licensing guidelines. The other license option may be to charge customers if they expect to benefit financially from their modifications, for example.
- Dual licenses are especially being considered by those developers who are experiencing customers modifying their tools and profiting off these modifications.

Least used ■ ■ ■ □ Most used

Fundraise / crowdfund

Definition: Developer solicits donations or sells branded merchandise to support ongoing maintenance of the digital tool or to develop a new feature.

Note: With a fundraising or crowdfunding model, developers solicit small contributions from many contributors. This differs from donor or grant funding which includes large contributions from a few contributors.

Benefits	Challenges
• Provides visibility to the broader public for a digital tool. • Targets can be set based on the needs of the tool.	• Involves high costs (e.g., staffing costs, marketing costs, etc.) for developers. • Contributors may expect compensation from the developer in return for their donations, such as free access to the digital tool. • Success is hard to predict.

Considerations for global goods

- This model has been used by free and open-source software developers like Wikipedia and Mozilla that operate outside of digital health.
- Like donor funding, this model has been suitable for developers in need of capital for development costs, but it is less successful once a digital tool has been developed.
- No developers mentioned using this model to finance ongoing *development* and *maintenance* (i.e., operating costs) and was typically viewed as a supplementary financing stream.

Least used ■ □ □ □ Most used

Gain sharing

Definition: Customer agrees to pay a portion of the monetary gains that come from implementation of the digital tool

Benefits	Challenges
• Ensures developers receive compensation when customers are paid to implement their digital tool. • Grows revenue proportionally with the scale of implementation. • Does not require the developer to possess implementation capabilities	• Requires negotiations between customers and developers to determine payment frameworks. • Is difficult to enforce for developers of digital tools with an open source license.

Considerations for global goods

- Another model that aligns with gain sharing is a pay-for-performance model. In digital health, this may be based on measured improvements to health outcomes. It is unknown if customers would be interested in pay-for-performance but some developers are considering this model.
- Developers that have implementation capabilities highlighted that they were often outbid by other implementers. In these cases, developers are still responsible for maintenance of the digital tool but are not compensated through the implementation grant.
- Developers can enforce gain sharing using a dual license model to earn a portion of the monetary gains from the implementation of the digital tool.

Least used ■ □ □ □ Most used

Microtask

Definition: User receives access to digital tool in exchange for completing a “microtask” that the developer can monetize (e.g., to access the tool, the user must identify objects in an image, with the output being sold to train machine learning models).

Benefits	Challenges
• Can be used to train artificial intelligence models, which may lead to many customers.	• Relies on high numbers of users to generate outputs that can be monetized. Customers may target digital tools with high user traffic.

Considerations for global goods

- Companies willing to pay for microtasking would most likely pay small amounts on a per use basis, limiting the revenue potential of this model. Even for digital tools with high traffic, microtasking is likely only a supplementary revenue model.
- No data was provided in interviews on this model.

Least used ■ ■ ■ □ Most used

Software as a service (SaaS)

Definition: Customer pays fees to access cloud-based digital tool—usually through a subscription or pay-per-use.

Note: SaaS is typically offered through a subscription service, which can include several services such as cloud hosting, software maintenance, and helpdesk support. This differs from supporting services, where services can be purchased separately.

Benefits	Challenges
• Offers a predictable source of revenue—especially for digital tools beyond the development stage. • Useful for paying for ongoing operating expenses which are high. • Adapts based on customer need and is suitable for a variety of customer types.	• Involves cloud hosting by the developer, making the developer responsible for cybersecurity and maintaining compliance with local regulations. • May be limited to specific countries if on-premises hosting is required.

Considerations for global goods

- Developers must offer a value proposition that exceeds the benefits of using an open-source digital tool independently. As hosting is the developer's responsibility, the total cost for a customer is often less than to self-hosting.
- More customers are seeing the benefits of delegating hosting responsibilities, decreasing the resistance to SaaS.
- Respondents are enthusiastic about this model due to its potential revenue benefits and are working to address barriers to employing the model, particularly those related to cybersecurity.

Least used ■ ■ ■ ■ Most used

Supporting service

Definition: Customer pays for complementary services to digital tool (e.g., training, data analysis, data visualization).

Benefits	Challenges
• Provides revenue beyond the development stage. • Can be offered at multiple tiers. Tiered support allows developers to provide support based on customer capacity. • Resources (e.g., training materials and data analysis tools) can be recycled and tailored for other customers. • Allows customers to maximize the benefits a tool can provide.	• Increases operational costs for developers, as providing these services often requires training and investment in infrastructure, such as data visualization software. • Requires human resources to employ. Other models like SaaS model require fewer human resources. • Is difficult to scale given the operational costs and human resources needed to use the model.

Supporting service (cont.)

Considerations for global goods

- National or regional implementations offer large contracts. Developers are increasingly considering how to be more involved with the implementation.
- Customers requesting supporting services emphasize technical assistance (e.g., helpdesk support and training) and project management. Developers acknowledged that nearly all customers need supporting services to some extent.
- Variable or usage-based models (e.g., per user, per facility, etc.) have been used to determine pricing for supportive services.
- Service agreements can be billed hourly, monthly, or annually—based on the level of support needed.
- The flexibility of this model, with its adjustable levels of support and pricing, makes it a popular choice for both customers and developers.
- A key challenge for developers is that customers often underestimate the costs associated with implementing supporting services.
- This model is a primary revenue generation tactic for many developers.

Least used ■ □ □ □ Most used

White label

Definition: Customer pays for unbranded digital tool—allowing another business to use it under their own brand name.

Benefits	Challenges
• Can be employed simultaneously with other models. • Reduces costs for customers by providing a pre-built tool, eliminating the need to develop one from scratch. • Allows the same product to cater to a range of customers.	• Mostly used in e-commerce and IT sectors–not digital health. • Requires a tool that customers are seeking to rebrand as their own. • Customers may not see the need to rebrand as critical, decreasing willingness to pay.

Considerations for global goods

- Certain types of digital tools may have a customer base more interested in a white label model than others (i.e., willingness to rebrand an electronic medical registry that allows patient access, but not a logistics management information system).
- A white label model can be used in conjunction with a SaaS revenue model, a freemium pricing model, or a Robin Hood pricing model. An example would be charging customers in high-income countries a fee but not in low-income countries.

Least used ■ □ □ □ Most used

Freemium

Definition: Developer offers a “basic” version of the digital tool for free, and a “premium” version for a fee.

Benefits	Challenges
• Can offer the digital tool to customers with varying capability to pay. • Generates revenue from customers willing to pay to for more features.	• Requires converting “basic” users to “premium” users, which can be difficult. • Does not guarantee developers will recover the costs of development. • Needs adequate number of customers to maintain the “premium” version.

Considerations for global goods

- Developers committed to following global goods standards are concerned that Freemium restricts user access and are reluctant to use this model.
- Developers open to selling a “premium” version of their tool say it is difficult to determine what capabilities to provide for free and what should be paid for. Providing enough functionality to users to encourage them to buy premium features is difficult in practice.

Least used ■ ■ ■ □ Most used

Robin Hood

Definition: Developer charges a different price for the digital tool based on ability-to-pay (e.g., different price for customers in high-income countries vs. low-income countries).

Benefits	Challenges
• Allows customers who have the financial means to subsidize the costs for customers who do not. • Can be a potential revenue stream from non-traditional funders.	• Maybe more suited for digital tools that are designed for use in both high- and low-income settings.

Considerations for global goods

- The Robin Hood model can apply to fees for supporting services or customization rather than the digital tool itself for global goods developers committed to remaining fully open-source.
- Developers have also used revenue generated from customization to fund pro-bono small-scale implementations. These pro-bono implementations benefit both users and developers, offering an opportunity to test tool modifications.
- Many global goods are designed for users in low-income settings. Product attributes like offline functionality, short message service (SMS) capability, and data compression are more compelling to customers in low-income settings and may be less enticing to customers in high-income settings.
- Developers have explored ways to limit free access to their tools for customers in high-income countries. One method could be a dual-license model based on where the tool is accessed.

Least used ■ ■ □ □ Most used

Variable or unit-based model

Definition: Customer pays for use of the digital tool based on number of units utilized during a specified duration (e.g., users per month).

Benefits	Challenges
• Determines price based on adjustable variables or units. • Offers flexibility to customers, who adjust their usage based on their budget and needs. • Can decrease prices for customers due to economies of scale as tools contract more and larger customers.	• Is difficult to implement due to billing complexity. • May incentivize sharing passwords for *per user* unit-based pricing—introducing a potential security risk.

Considerations for global goods

- Developers use this model frequently when determining the pricing for supporting services and customization.
- Developers often determine price based on the number of facilities where the tool is implemented.

Least used ■ ■ ■ ■ Most used

Community support

Definition: A community of users supports the development, maintenance, and/or development of new features for the digital tool.

Benefits	Challenges
• Model creates an ecosystem where the users act as contributors to the core digital tool by developing new features and improving or expanding capabilities. • Can make a digital tool more sustainable if there is a broad set of contributors that have a stake in the success of the digital tool. • May require fewer paid human resources if community users support core functions. • Allows users to customize a tool to meet their needs.	• Requires a strong management team. • Relies on community users with other responsibilities, which leads to infrequent contributions. • May be unreliable for addressing urgent updates or maintenance.

Community support (cont.)

Considerations for global goods

- The community support model varies greatly amongst global goods developers, but all global goods developers surveyed relied on community support to some extent—ranging from direct contribution for development and maintenance to input on specific use cases.
- Respondents who have had success with community support has a core team that manages contributions. Core team responsibilities include quality assurance on community contributions, maintaining involvement of the community, and handling disagreements about the strategic direction.
- Some developers discussed moving to a decentralized approach where there is not a core team to manage community contributions. No respondents have committed to this organizational structure.

A variety of models are available to global goods software developers

- **Donor funding remains the core model** for funding digital health software global goods—indicating an ongoing market inefficiency. Additionally, most donor funding goes to *implementation*, creating an additional gap in funding for ongoing *development* and *maintenance* (i.e., operating costs).
- Each revenue, pricing, and cost model was **considered by at least one digital health software developer**.
- Supporting service and customization models are the most common models used to supplement donor funding. These models are scalable and are **well-suited to global goods** since they do not restrict free and open-source software requirements.
- While this analysis highlights distinct models, the **models often overlap in practice**—providing developers with a spectrum of diverse options (e.g., a developer who offers SaaS could lease server space for other uses by applying the core capability model).
- Strong **ownership by the user** (e.g., ministry of health) ensures the tool is well-suited to the health system and improves opportunities for funding—either directly from the user or from donors. Customization, supporting service, and white label models are well-suited to supporting user ownership.
- Developers should consider how some revenue models (e.g., dual license, SaaS, white label) may interact with free and open-source standards. These models may limit community support.
- Advertisement / sponsorship, data monetization, and microtasks may not be appropriate for the global health context due to **potential ethical concerns**.
- Some models can be **resource intensive** to support and may require developers to directly implement the software.

Many factors need to be considered when a developer decides to directly implement their tool

Directly implement the software		Let others implement the software
Number of models available to augment donor funding		**Number of models available to augment donor funding**
A variety of revenue, cost, and pricing models are available for those that implement their software.	Favors Implementation	Only select non-donor funded models (e.g., certification, crowdfunding, supporting services, community support) are feasible for those not involved in implementation, challenging financial sustainability.
Understanding user needs		**Understanding user needs**
Software developers that also implement their solutions in a variety of geographical contexts may not have the same understanding of user needs.	Favors partnering	Partnering with local implementers for implementation is promising and supports local capacity building.
Ability to support a broad community of users		**Ability to support a broad community of users**
Customizations from project implementations can enhance the core software if integrated, but it's important to ensure these updates align with the broader community's needs.	Mixed	Software developers that do not implement can focus on developing a product that is acceptable to the broad needs of the community rather than a specific project.

Assumptions and limitations

- The literature review did not include a comprehensive search for ‘pricing’ or ‘cost’ models. These findings may have gaps in understanding for those models.
- The model list focused on models that could allow global goods to remain free and open-source, so it is not a comprehensive list of all financial models.
- The survey and interviews may not be representative of all digital tool developers. The survey was voluntary, and interviewees were selected based on availability and willingness to speak. Most interviews by design focused on global goods operating in low resource settings.
- While definitions were provided to each respondent, familiarity with each model may have varied and usage ratings should be viewed as an estimate.
- Most respondents were global goods software developers, and the results may be less relevant to proprietary software developers working in low-resource settings.

Digital Square is supported by:

Digital Square is a PATH-led initiative funded and designed by the United States Agency for International Development, the Bill & Melinda Gates Foundation, and a consortium of other donors.

This presentation was made possible by the generous support of the American people through the United States Agency for International Development. The contents are the responsibility of PATH and do not necessarily reflect the views of USAID or the United States Government.

https://digitalsquare.org

Digital Square is a PATH-led initiative funded and designed by the
United States Agency for International Development,
the Bill & Melinda Gates Foundation, and a consortium of other donors.