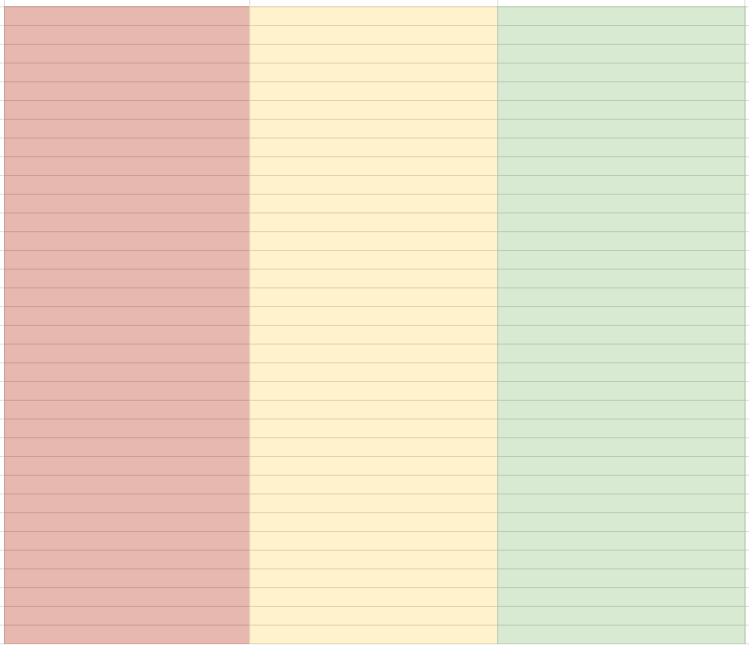
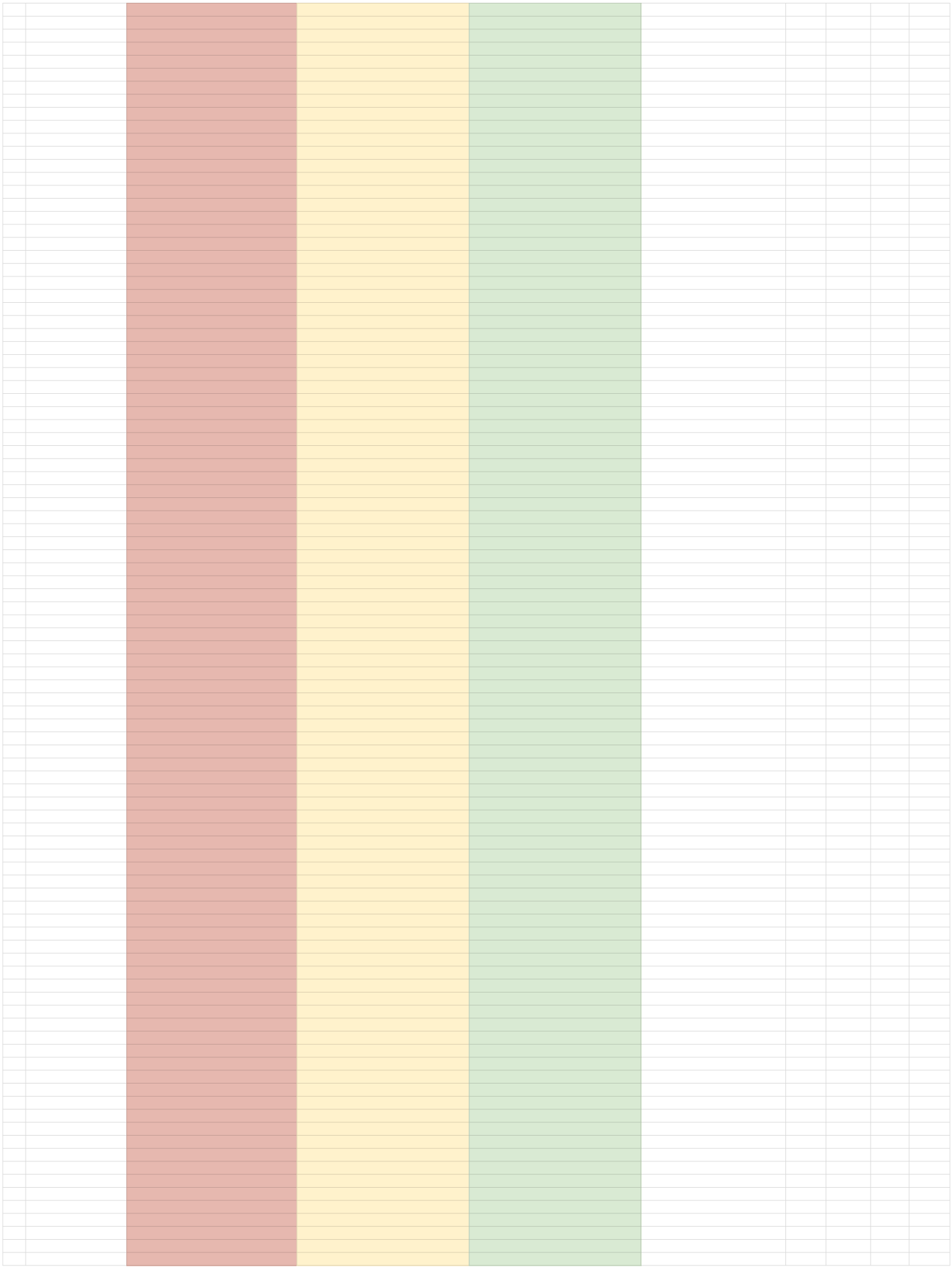
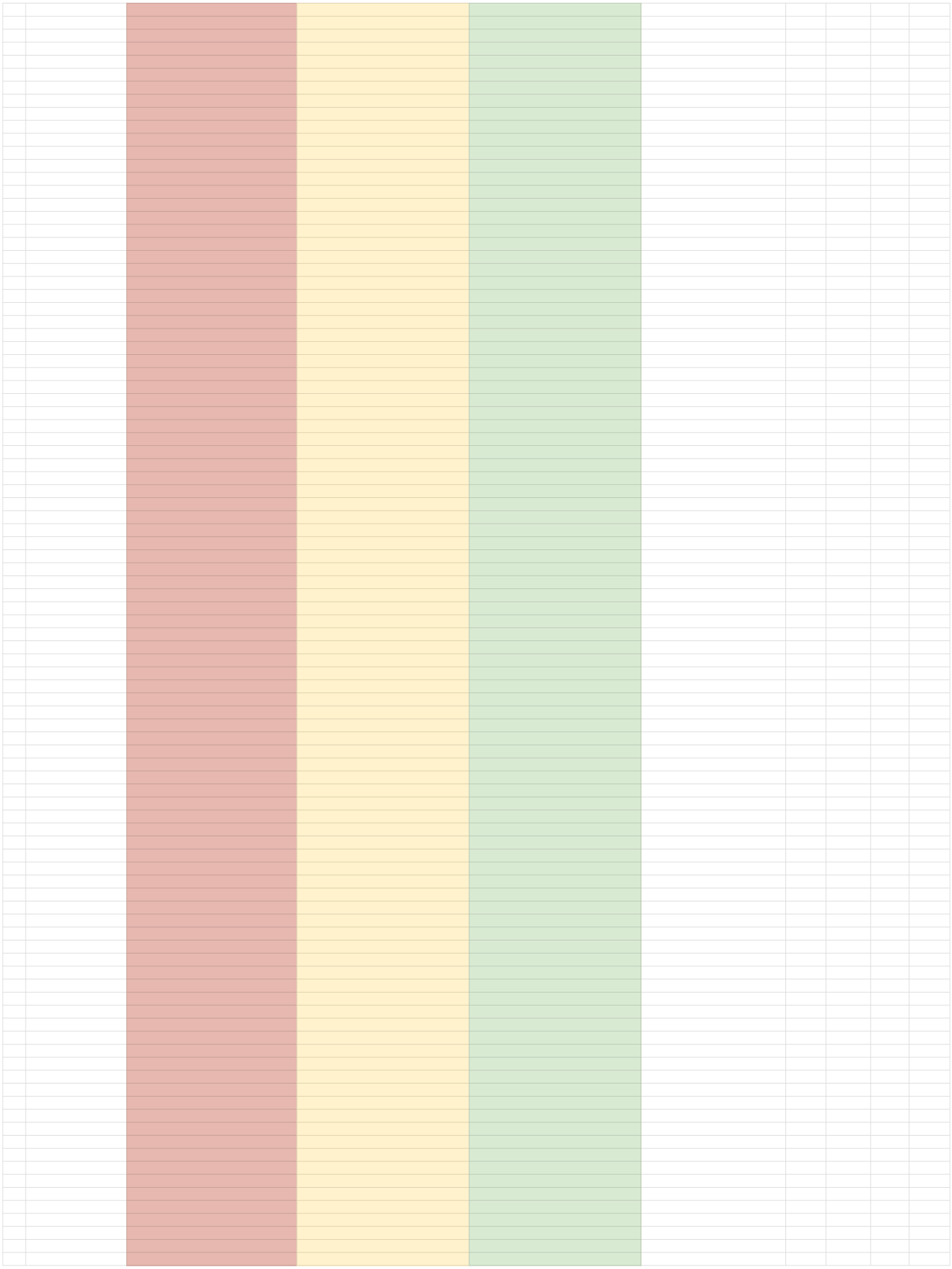
Global Good Maturity Model for digital health software tools.

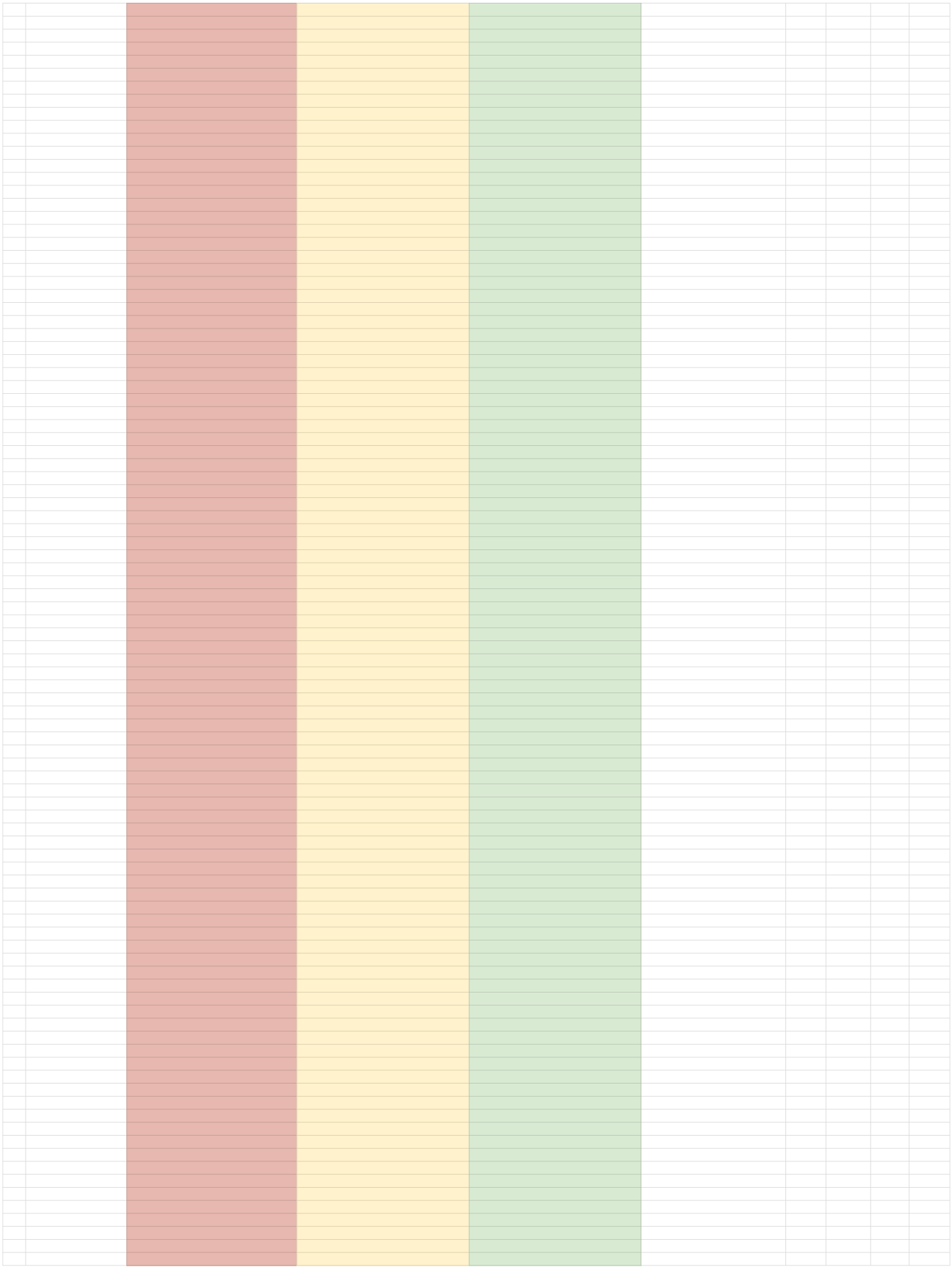
Version 1.0

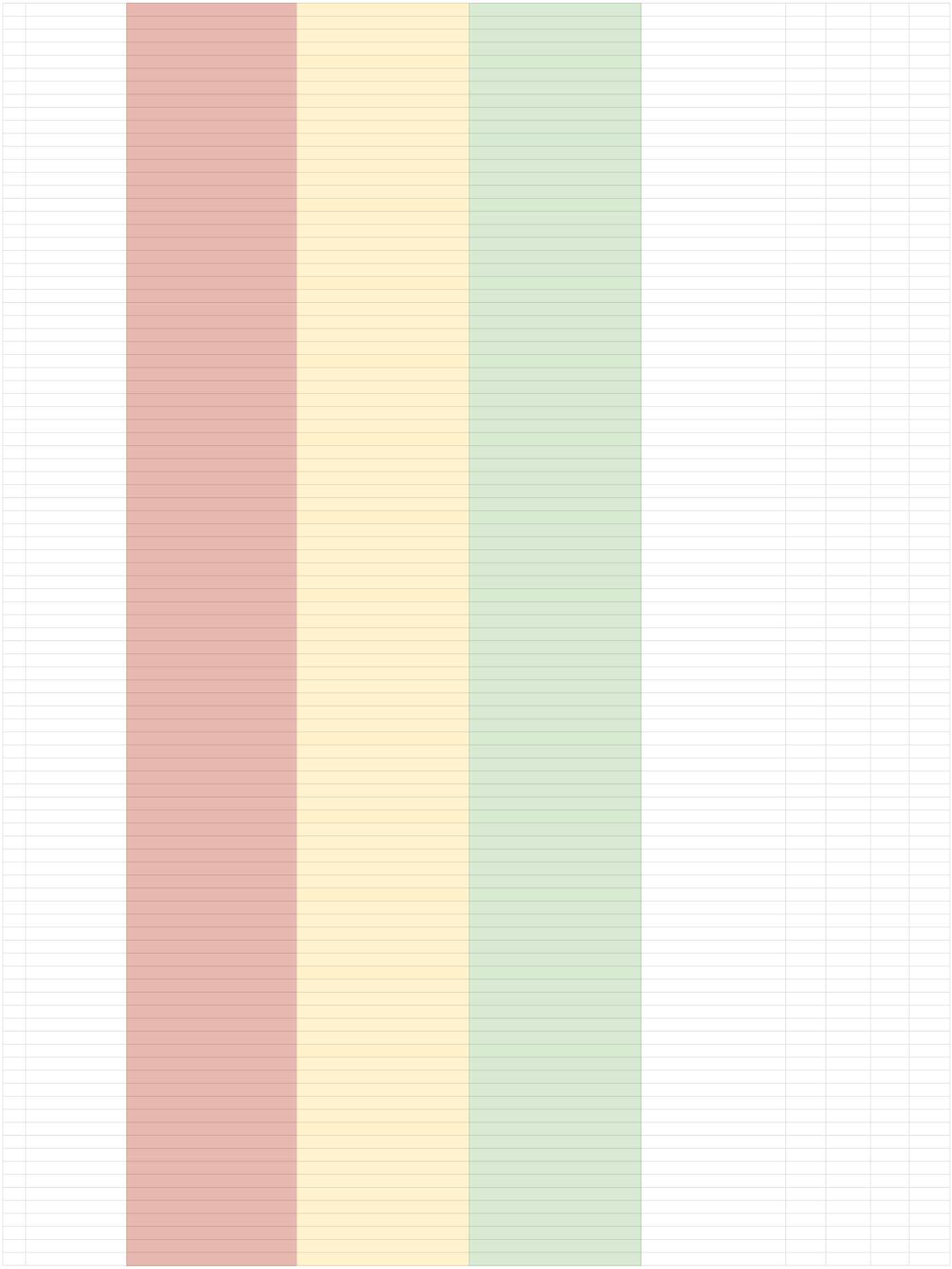
core indicator	sub-indicator	Low	Medium	High	Notes
Global Utility	**Country Utilization**	Less than two countries or states actively use the tool for use as part of their health information system	At least four countries or states actively use the tool for use as part of their health information system with at least 20% of total nation-wide or state-wide target users routinely using product/service as intended	At least ten countries or states actively use the tool for use as part of their health information system with at least 30% of total nation-wide or state-wide target users routinely using product/service as intended	
	Country Strategy	Less than two countries or states have included the tool as part of their eHealth strategy or framework	At least four countries or states have included the tool as part of their eHealth strategy or framework	At least ten countries or states have included the tool as part of their eHealth strategy or framework	
	Digital Health Interventions	the tool does not meet digital functional requirements (as defined by WHO's Classification of Digital Health Interventions) without signifigant customization or configuration	the tool does partially meets digital functional requirements (as defined by WHO's Classification of Digital Health Interventions) without signifigant customization or configuration	the tool does fully meets digital functional requirements (as defined by WHO's Classification of Digital Health Interventions) without signifigant customization or configuration	
	Source Code Accessibility	source code not publically available or not released under an open-source license	source code exists on a publicly accessible repository and licensed under an Open Source Initiative approved license.	source code exists on a publicly accessible repository and licensed under an Open Source Initiative approved license. Software is structured to allow local customizations and new modules and functionality without requiring forking of main code	
	Funding and Revenue	at most two revenue streams exists. revenue streams are largely dependent on time bound project implementations	multiple revenue streams/funders exist across project implementations	multiple revenue streams and funding mechanisms exist including at least one that provides for multi-year support of core software development, documentation and other key artifacts.	a revenue stream indicates a source of funding to support the development of a global good. such revenue streams could come from donor contributions, from one of the variety of business models used by open source software tools to fund their continued development, or from in-kind contribution from an organization
Community Support	**Developer, Contributor and Implementor Community Engagement**	Less than 10% of estimated total of developers, contributors and implementers are on a communication platform	Up to 20% of estimated total of developers, contributors or implementers, including some country representation, are engaged on a communication platform.	At least 30% of estimated total developers, contributors and implementers are engaged on a communication platform. community leadership includes representation from countries where the tool is deployed	
	Community Governance	there is no community governance structure in place to direct continued development of the digital health tool	some informal processes for community management exist to direct continued development of the digital health tool	formal community structures (e.g. leadership, technical advisory group, community representatives) exist and are practiced with documented roles and responsibilities in a transparent fashion and are used to direct continued development of the digital health tool	
	Software Roadmap	no software roadmap exists or there is no publicly accessible and routintely maintained platform for new feature requests	there is a publicly accessible and routintely maintained platform for new feature requests. a software roadmap exists describing currently planned and resourced development activities	new features and functionality are documented as part of a software roadmap as part of a release cycle. there are forums for community members to discuss new feature requests. a clear prioritization process exists and is utilized for the development of new features and functionality as part of a product backlog.	
	User Documentation	no user documentation exists	some user documentation exists (training manual, demo videos) but only addresses a limited subset of common functionality	a full suite of user documentation exists including training manuals, online courses, tutorials and implementation guides addressing most of the common functionality. documentation has been released under a Creative Commons license	
	Multi-Lingual Support	Limited or no support in the software for multiple languages. Multi-lingual documentation / user resources are practically non-existent	Software has be internationalized to support multiple languages (though may not have been translated) for primary portions of the user interface. Some user documentation exists in more than one language	Software has been translated into multiple languages and fully supports internationalization requirements. There is an easy tool for new translations to be added. Significant parts of user and implementer documentation has been translated into at least one other language.	
Software Maturity	**Technical Documentation**	no substantial documentation of the software exists	some technical documentation exists of the source code, use cases and functional requirements	source code is documented to the point that new adopters can customize and add new functionality with relying on significant help from one of the core developers. online courses or tutorials are available to address common development and deployment tasks. core business workflows and functional requirements are fully documented using use cases, user stories or other equivalent methodology	
	Software Productization	no documentation available for deployment and configuration	full documentation available for deployment and configuration. a new implementation does not require the involvement of the core development team	software has been packaged for one or more common operating systems or platforms. software upgrades can largely be achieved without manual intervention. unit or integration testing is part of the release process.	
	Interoperability and Data Accessibility	extract or importing data into the system usually requires looking at source code and/or directly accessing database	some APIs are available for accessing and managing data. there are user facing interfaces to export core data and metadata in the system (e.g. in CSV format) for further analysis and data transfer purposes	a robust API is available for key data and metadata exchange needs for the primary business domain with functional requirements for the API having been developed in conjuction with appropriate country, regional and global stakeholders. API endpoints exist for core data and metadata elements which adhere to standards developed by an appropriate Standards Development Organization relevant to the tools business domain. standards based API endpoints are used in at least four jurisdictions (e.g. countries or states).	
	Security	No security controls or implementation guidance is in place.	Role based authorization exists, if appropriate.	Role based authorization exists, if appropriate. All remote access (web interface, APIs) are encrypted by default using current best practices. An independent security audit of the software has taken place within the last twelve months.	
	Scalability	There are no jurisdicions (e.g. country, state) that manage 10% of their "entities" within the tool and no performance and load statistics exist.	There is at least one jurisdicion (e.g. country, state) deployment for which 20% of all "entities" are managed within the software. There has been at least one evaluation of software performance / load testing	There is at least one jurisdicions (e.g. country, state) deployment for which 30% of all "entities" are managed within the software. Performance and load testing is a part of routine releases and results are publicly available.	Entities are the data objects that are central to the pimary business domain that the software addresses. For example, an EMR would have a patient as one of its entities.

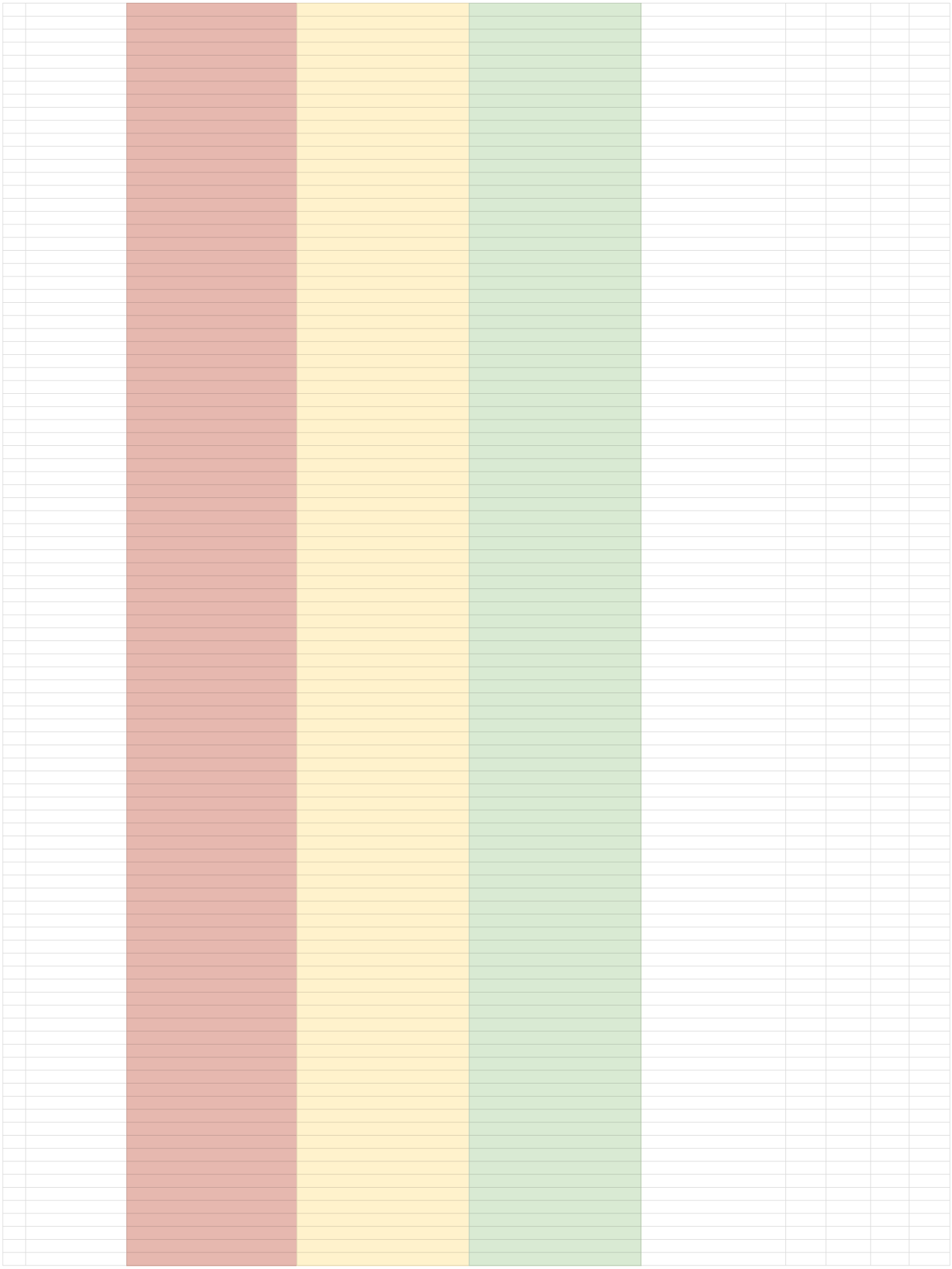


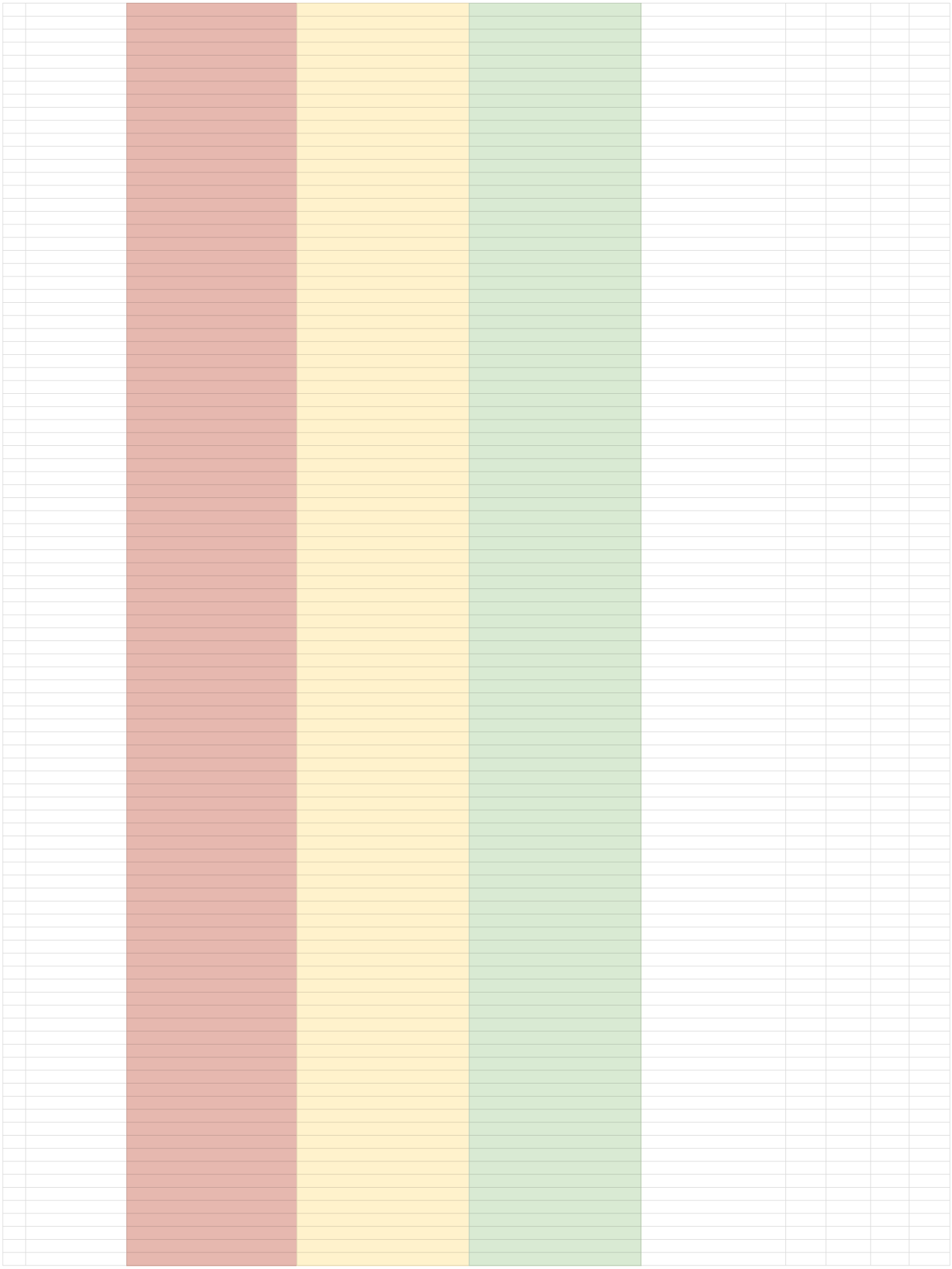


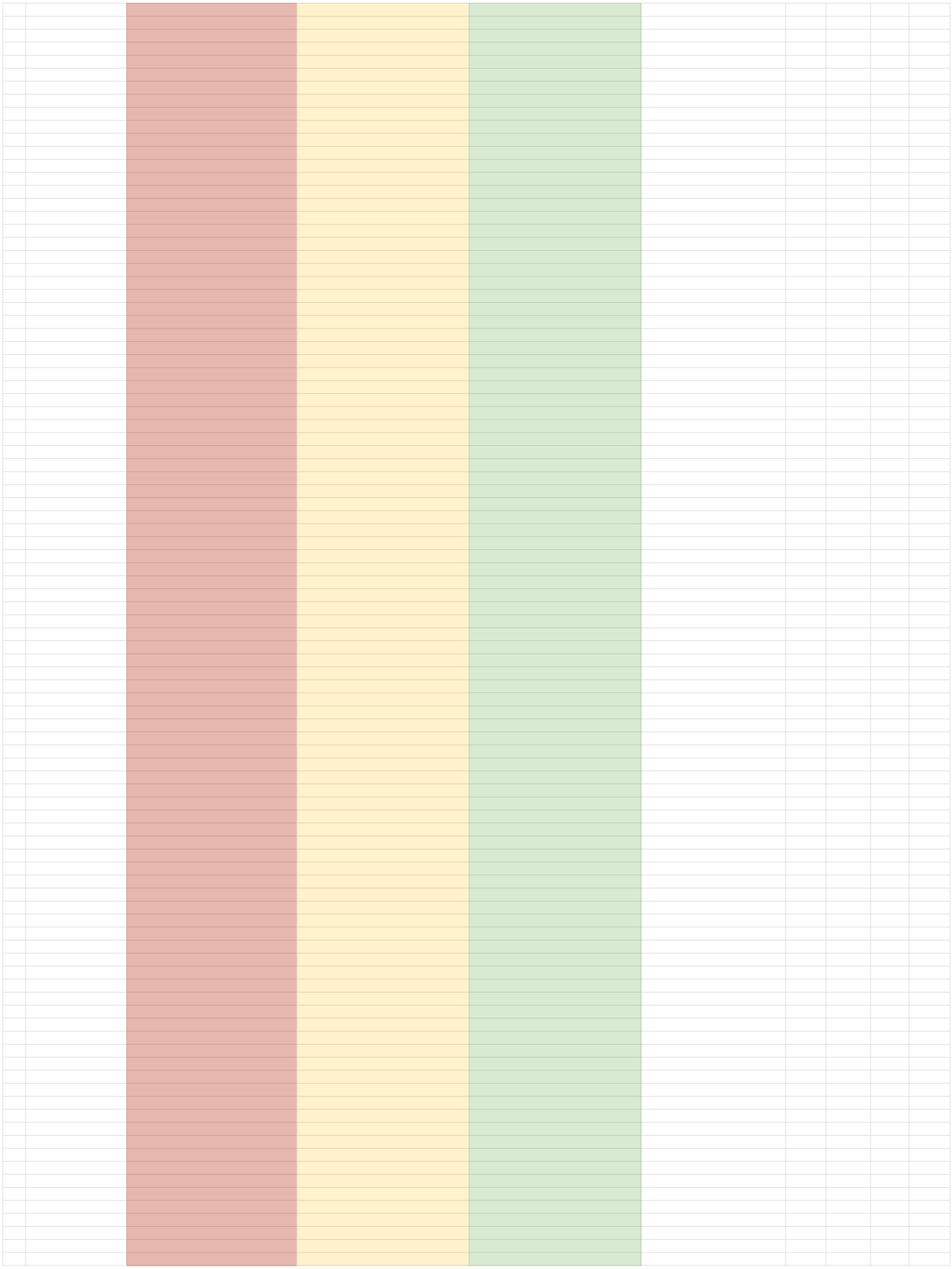












Example Rating of a Digital Health Software Global Good
(make a copy of this document to use)

Core Indicator and Calculated Score [0-10]	Sub-Indicator	*change rating here*		Notes on Core Indicators for OpenLMIS
Global Utility	Country Utilization	Medium	At least four countries or states actively use the tool for use as part of their health information system with at least 20% of total nation-wide or state-wide target users routinely using product/service as intended	OpenLMIS is currently being used to manage health commodity logistics processes at over 10,000 health facilities in eight geographies: Benin, Cote d'Ivoire, Guinea, Malawi, Mozambique, Tanzania, Zanzibar, Zambia. Of these, 3/8 implementations are nationwide and cover all health facilities in the country.
6	Country Strategy	Medium	At least four countries or states have included the tool as part of their eHealth strategy or framework	All countries currently using OpenLMIS have included the tool as part of the eHealth framework, and additional countries (notably Angola and DRC), are including it in their eHealth strategy. In 2019, we anticpate OpenLMIS being rolled out and expanding its use within Angola, Mozambique and potentially Benin.
	Digital Health Interventions	High	the tool does fully meets digital functional requirements (as defined by WHO's Classification of Digital Health Interventions) without signifigant customization or configuration	OpenLMIS falls most clearly under category 3.2, Supply Chain Management, and fully meets the digital functional requirements of this category, either through functionality available in the system or through integration/interoperability with another system.
	Source Code Accessibility	High	source code exists on a publicly accessible repository and licensed under an Open Source Initiative approved license. Software is structured to allow local customizations and new modules and functionality without requiring forking of main code	-All source code for OpenLMIS publicly available on GitHub -Project documentation publicly available on Confluence and ReadtheDocs -Project ticket tracking publicly available in JIRA -OpenLMIS operating under GNU Affero open source license -Software architecture is modular, micro-services based to facilitate customization without forking
	Funding and Revenue	Low	at most two revenue streams exists. revenue streams are largely dependent on time bound project implementations	OpenLMIS currently receives funding from two donors.
Community	Developer, Contributor and Implementor Community Engagement	Medium	Up to 20% of estimated total of developers, contributors or implementers, including some country representation, are engaged on a communication platform.	Nearly all (100%) of developers on OpenLMIS are engaged with and utilizing several communication platforms; however, in-country users are underrepresented in community leadership.
7	Community Governance	High	formal community structures (e.g. leadership, technical advisory group, community representatives) exist and are practiced with documented roles and responsibilities in a transparent fashion and are used to direct continued development of the digital health tool	The OpenLMIS Community comprises three leadership committees. All notes and recordings from Committee meetings are publicly available on the OpenLMIS wiki. -Governance Committee: Directs overall initiative goals and provides leadership -Product Committee: Discusses and approves product features -Technical (Dev) Committee: Discusses technical questions relating to software development and architecture
	Software Roadmap	High	new features and functionality are documented as part of a software roadmap as part of a release cycle. there are forums for community members to discuss new feature requests. a clear prioritization process exists and is utilized for the development of new features and functionality as part of a product backlog.	All functional documentation in OpenLMIS is publicly available on a variety of platforms and is tracked/described in the OpenLMIS Living Product Roadmap on the OpenLMIS wiki. Features are released as part of a regular release cycle and new feature requests are submitted and discussed via the OpenLMIS Product Committee.
	User Documentation	Medium	some user documentation exists (training manual, demo videos) but only addresses a limited subset of common functionality	-A small number of demo videos are available on the OpenLMIS YouTube Channel -A configuration guide is available -An initial version of the Implementer Toolkit is available
	Multi-Lingual Support	Medium	Software has be internationalized to support multiple languages (though may not have been translated) for primary portions of the user interface. Some user documentation exists in more than one language	- Most of the system has been translated into 3 languages (French, Portuguese, and English). - The software fully supports internationalization requirements so contributors can provide translations and implementers can override translations. - Currently, most user documentation is only in English.
Software	Technical Documentation	High	source code is documented to the point that new adopters can customize and add new functionality with relying on significant help from one of the core developers. online courses or tutorials are available to address common development and deployment tasks. core business workflows and functional requirements are fully documented using use cases, user stories or other equivalent methodology	- Technical documentation exists for the source code and APIs from each micro-service. - Use Cases and Personas are used in Jira tickets to define each feature as it is built. - Technical documentation exists; Implementer Toolkit, Configuration Guide, and Developer Onboarding Guide are available. - No online courses or tutorials are available for OpenLMIS.

8	Software Productization	High	software has been packaged for one or more common operating systems or platforms. software upgrades can largely be achieved without manual intervention. unit or integration testing is part of the release process.	- OpenLMIS is packaged with Docker to run on common operating systems and platforms, including Cloud providers. - OpenLMIS 3.x.y upgrades can largely be achieved without manual intervention (an exception is if the implementation has forked the UI in order to customize it). - Unit or integration testing is part of the release process. Thousands of unit and integration tests run automatically and manual test suites are also conducted for every release.
	Interoperability and Data Accessibility	High	a robust API is available for key data and metadata exchange needs for the primary business domain with functional requirements for the API having been developed in conjuction with appropriate country, regional and global stakeholders. API endpoints exist for core data and metadata elements which adhere to standards developed by an appropriate Standards Development Organization relevant to the tools business domain. standards based API endpoints are used in at least four jurisdictions (e.g. countries or states).	- OpenLMIS has robust RESTful APIs that are already leveraged by outside integrations and mobile apps (eg, SIGLUS mobile app in Mozambique using OpenLMIS REST APIs, or Nexleaf ColdTrace API integration using OpenLMIS FHIR Facility APIs). - OpenLMIS continues advocating for standards-based interoperability with other key players and standards including DHIS2, OpenSRP, GS1, OpenHIE, FHIR, etc.
	Security	Medium	Role based authorization exists, if appropriate. Guidance on encrypting all remote access (web interface, APIs) is available to implementors.	- Role based authorization exists, and all remote access (web interface, APIs) is encrypted by default using current best practices. - No independent security audit of the software has taken place. Significant security issues have been uncovered, reported and fixed, but other unknown flaws may still exist.
	Scalability	Medium	There is at least one jurisdicion (e.g. country, state) deployment for which 20% of all "entities" are managed within the software. There has been at least one evaluation of software performance / load testing	- Eight geographies have deployed OpenLMIS, including at national scale covering all health facilities and districts (with 90%+ reporting rates). - Performance and load testing happen manually and are not comprehensive. Results are shared openly but testing happens infrequently.
		Low		
		Medium		
		High		