



Swiss TPH



Swiss Tropical and Public Health Institute

CDSS Authoring

Patrick Delcroix - September 2025

CDSS Authoring

Agenda

- Added Value
- Authors
- Challenges
- Concepts
- Implementation

CDSS Authoring

CDSS Authoring Added Value

Transforms Static Guidelines into Dynamic Tools

Converts rigid paper-based guidelines into adaptable, digital formats.

Overcomes siloed organization and limited information capacity of paper.

Enables Local Customization

Adapts to site-specific needs, such as endemicity, local protocols, and drug availability.

Ensures relevance across diverse healthcare settings.

Clarifies Complex Calculations

Specifies algorithms and input parameters for accurate, automated decision-making.

Enhances precision in clinical recommendations.

Addresses Comorbidities & complex cases

Supports integrated management of multiple conditions.

Delivers holistic, patient-centered decision support.

CDSS Authoring

CDSS Authors

Subject matter experts (SME, e.g. Clinician) are the main author of a digital guideline, they are the one responsible of its content and therefore have the final word.

End users are providing requirements and feedback to ensure the the program will integrate as smoothly as possible in their daily tasks

IT specialist are supporting SME to indicate what is possible or not and are enriching the clinical content with technical elements to ensure that a solution could be generated out the authored content

Other stakeholders are supporting the SME making sure the content is fits the context (MoH, local implementor, Sponsors, ...)

CDSS Authoring Challenges

CDSS Authoring Challenges

Collaboration Between Stakeholders

- How to Bridge gaps between clinicians, MoH, Hospitals admin and IT teams ?
- How to have a clear responsibility over the content ?
- How to allow peer validation ?

➡ Diverse stakeholders need a clear, shared high-level view on the authored content.

CDSS Authoring Challenges

Representing Content for Intuitive Understanding

- How to have precise technical expressions yet not alienating non-technician ?
- How to have clinical expressions feel intuitive yet not ambiguous?
- How to ensure that implementation follow the authored content?
- How to avoid information overload that can hinder content understanding ?

➡ Authored content must balance technical precision with readability for non-IT specialists.

CDSS Authoring Challenges

Managing Treatments

- How to prevent double prescriptions in computed drug regimen?
- How to define diagnosis-specific drug alternatives without assuming per-drug equivalence?
- How to prioritize patient stabilisation in severe cases before referral?
- How to order recommendations alongside drugs for clarity and usability?

➡ Treatments demand decision trees for prioritization, selection, and safety, functioning as independent algorithms segment.

CDSS Authoring Challenges

Activity Management

- How to avoid fragmented designs with duplicate activities ?
- How to share activities within the guideline ?
- How to share activity across guidelines ?
- How to ensure activity have applicability rules when relevant ?

➡ Authored content need to be able to reuse the same activity definition in different context

CDSS Authoring Challenges

Sequence logic

- How to document conditionality to show only relevant information or tests?
- How to reuse task or activity without creating loops or dead ends?
- How to prevent premature stops from missing data required for treatment?
- How to ensure what follows an activity depend on the activity starting context

➡ Sequence logic needs advance capabilities to match clinician expectations

CDSS Authoring Challenges

Deciding When Content Should Start

- How to avoid running immunization before ETAT ?
- How to avoid Lab tests interrupting measurements activity, increasing cognitive burden ?
- How to avoid starting a guideline after the discharge message of another guideline ?
- How to prevent nice to have recommendations in emergency care case ?
- How to prevent “periodic evaluation” to load system if executed on untimely.
- How to avoid author guidelines as a overwhelming single documents ?

➡ Authored contents need to be splittable in different care “segments”

CDSS Authoring Challenges

Conversion

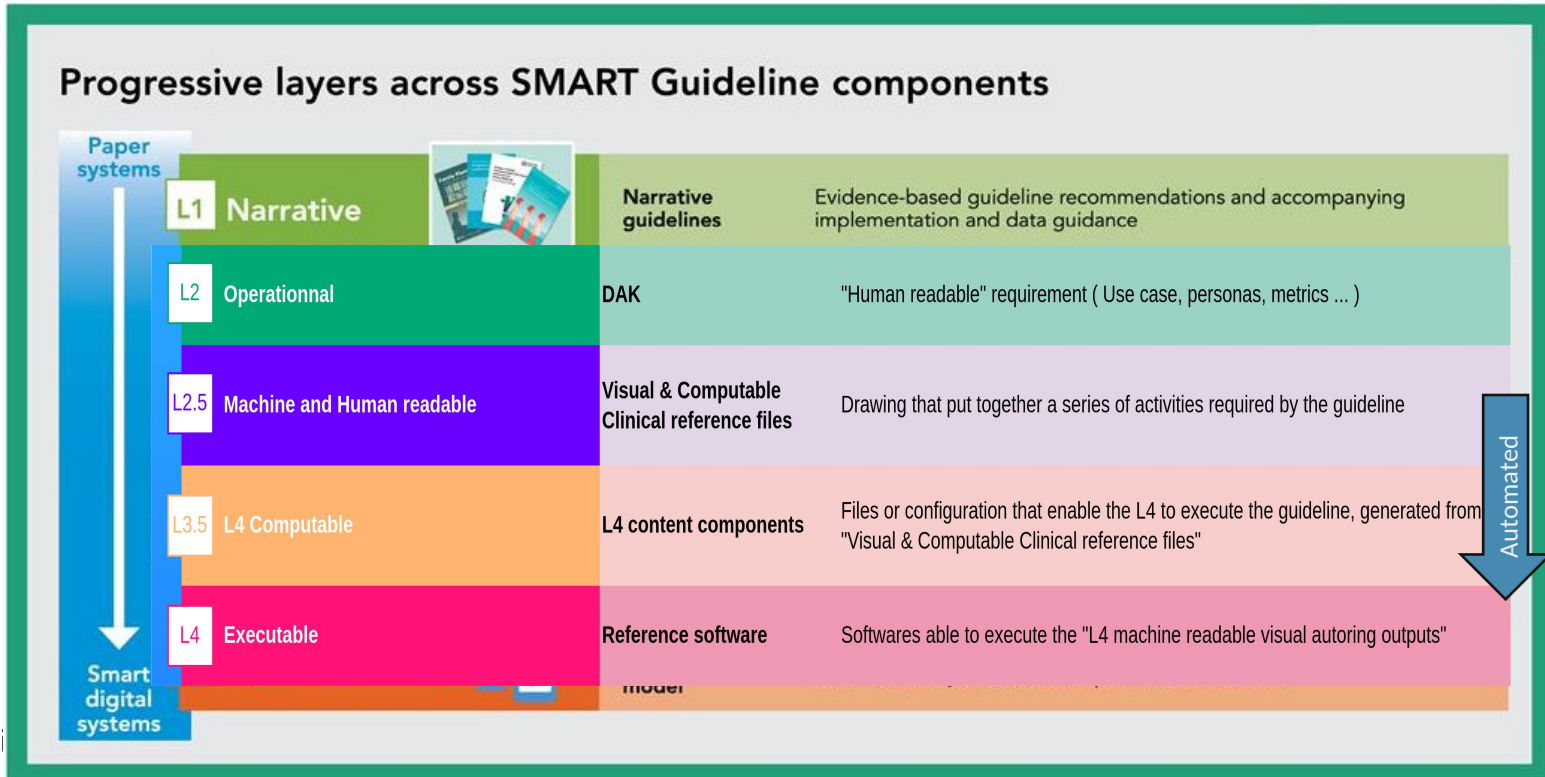
- How to avoid adding regression error on update ?
- How to ensure equivalence of same guideline on different L4?
- How to ensure the update is only changing the app in where the content was updated?
- How to prepare for integration and mapping ?

➡ Authored contents needs automatic conversion to prevent different kind of human errors

Visual Authoring Concepts

Visual Authoring

Mapping to WHO smart guideline layers



Visual Authoring Concepts

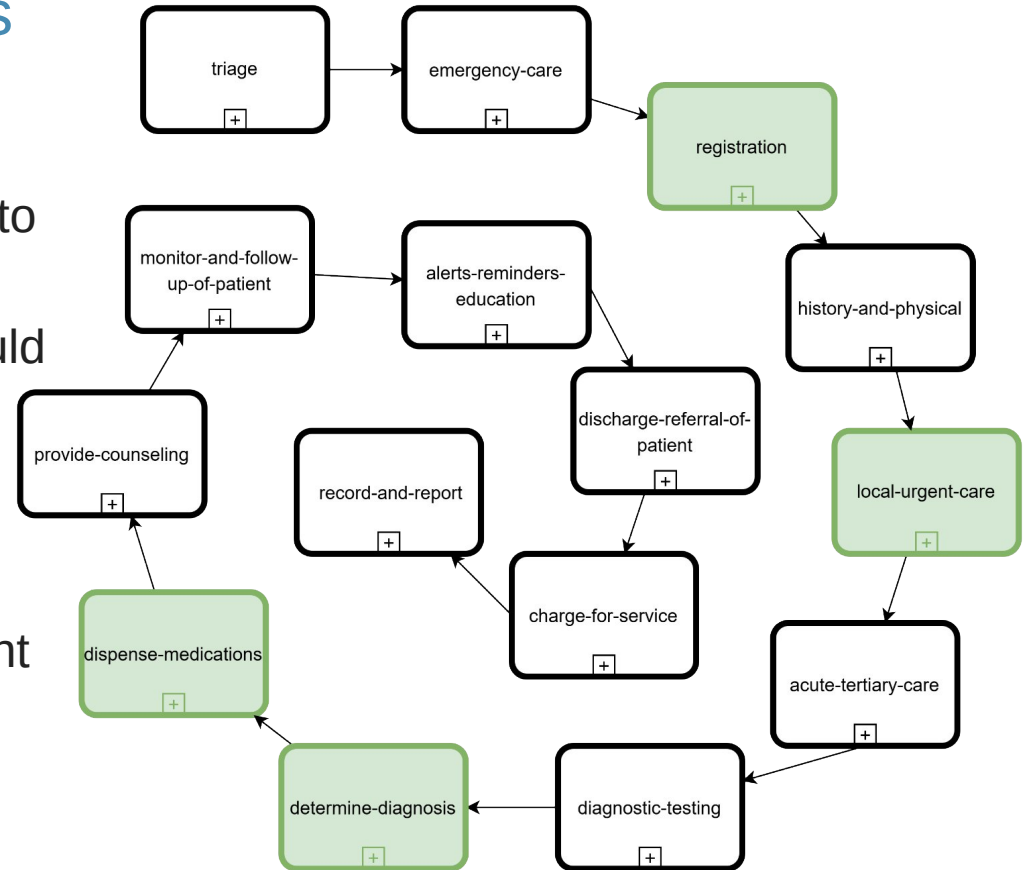
Overview

- **Purpose:**
 - What You See Is What You Get (WYSIWYG) for CDSS Authoring,
 - Share and reuse content
 - Deploy of different platforms
- **Concept:**
 - Define sequences with visual decision trees
 - Define logic with CQL or visual decision trees
 - Modular
- **Key advantage:**
 - Fully computable, enabling pipelines and rapid testing
 - Easier to make modification

Visual Authoring Concepts

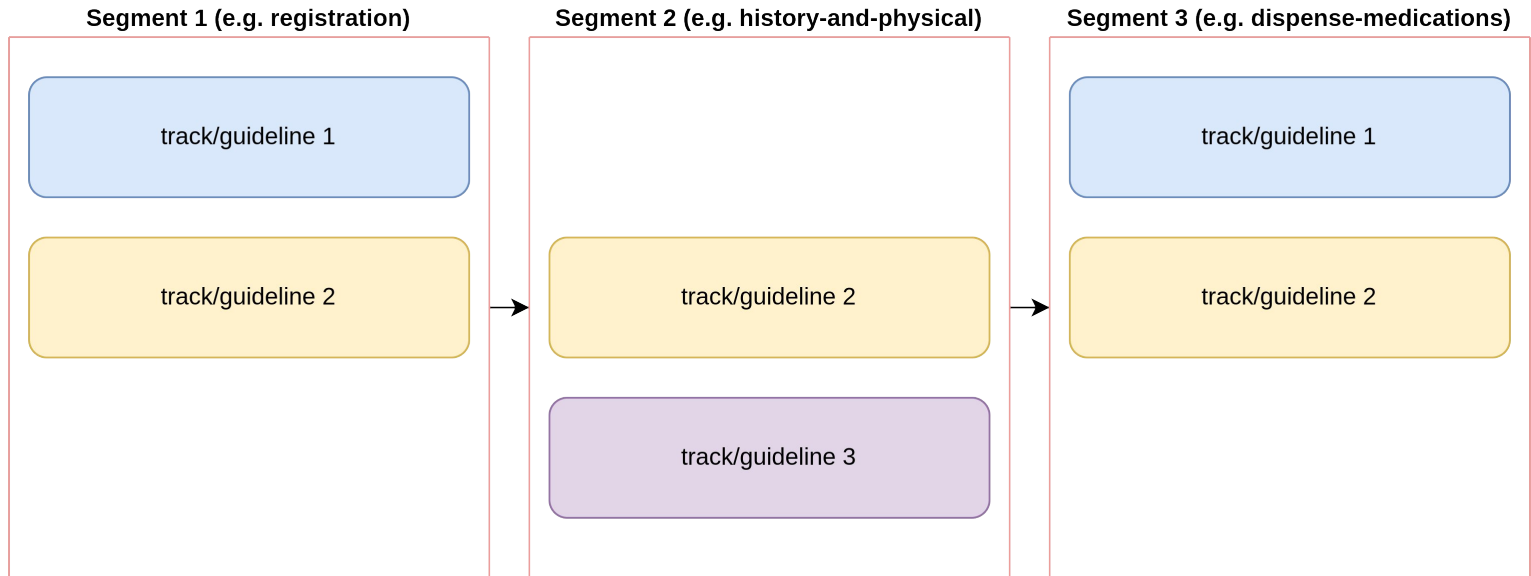
Segmented approach

- **Purpose:** divide the encounter into segments to:
 - Specify when an activity should be carried on
 - Merge different Guideline on segment level to create a coherent experience
 - Allow jump to a given segment



Visual Authoring Concepts

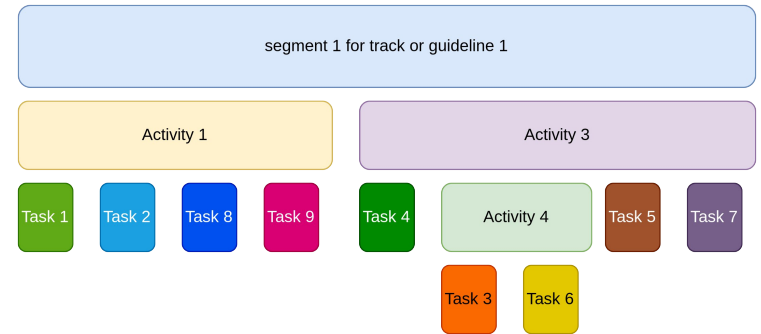
Segmented approach



Visual Authoring Concepts

Layered approach

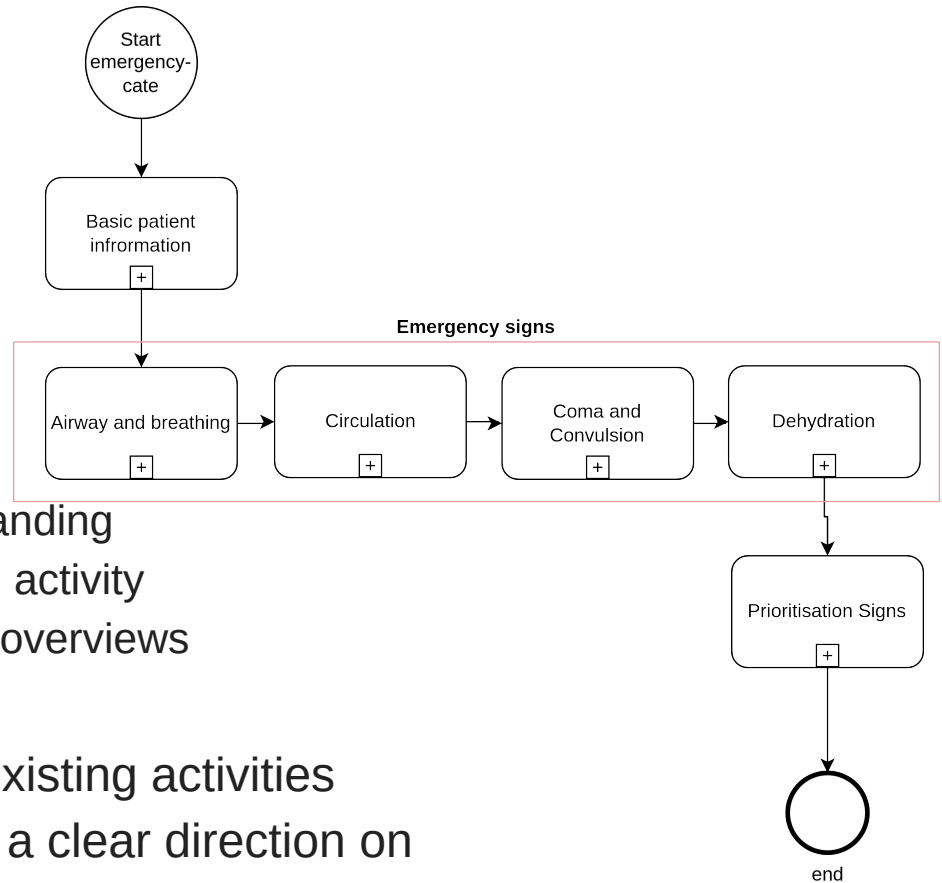
- **Purpose:**
 - Separation of the intent (WHAT) and the details (HOW)
- **Concept:**
 - For each process you want to address define 3+ levels:
 - An Segment overview that glue activities together
 - Activities that contains tasks but can also incorporate other activities
 - Node/Task within activities
 - Activities can be used in different overall / activities



Visual Authoring Concepts

Layer 1 : Segment overview

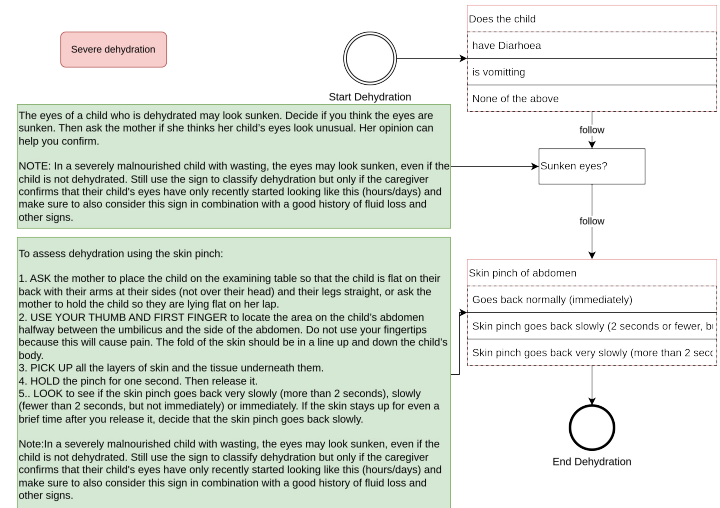
- **Purpose:**
 - Define WHAT need to be done
- **Concepts:**
 - Remain generic to ensure understanding
 - Each node (“box”) is detailed in an activity
 - Same segment can have several overviews
- **Key advantage:**
 - Segment overview can reuse existing activities
 - Segment overview should give a clear direction on what is addressed



Visual Authoring Concepts

Layer 2+ : Activity

- **Purpose:** Define HOW it need to be done
- **Concepts:**
 - An activity is a sequential decision tree
 - May have an **applicability** expression
 - “Done” if all end nodes are processed
- **Key advantage:**
 - Clear view how what the user will face
 - An activity could have several version based on the context or Patient
 - Same activity can be used in different guidelines



Visual Authoring Concepts

Layer 3+ : Nodes/tasks

- **Purpose:** Define a specific actions
- **Concepts:**
 - **Data capture nodes:** capturing patient data, findings
 - **Sequence node:** define how the logic is used, starts, ends ...
 - **Calculation node:** define complex logic using CQL
 - **Message nodes:** provide information to the user
- **Key advantage:**
 - Nodes type can be extended to use AI model, ...
 - May specify how it should be displayed on L4 (horizontal/vertical choice, scale ...)

Visual Authoring

How it helps

Visual Authoring

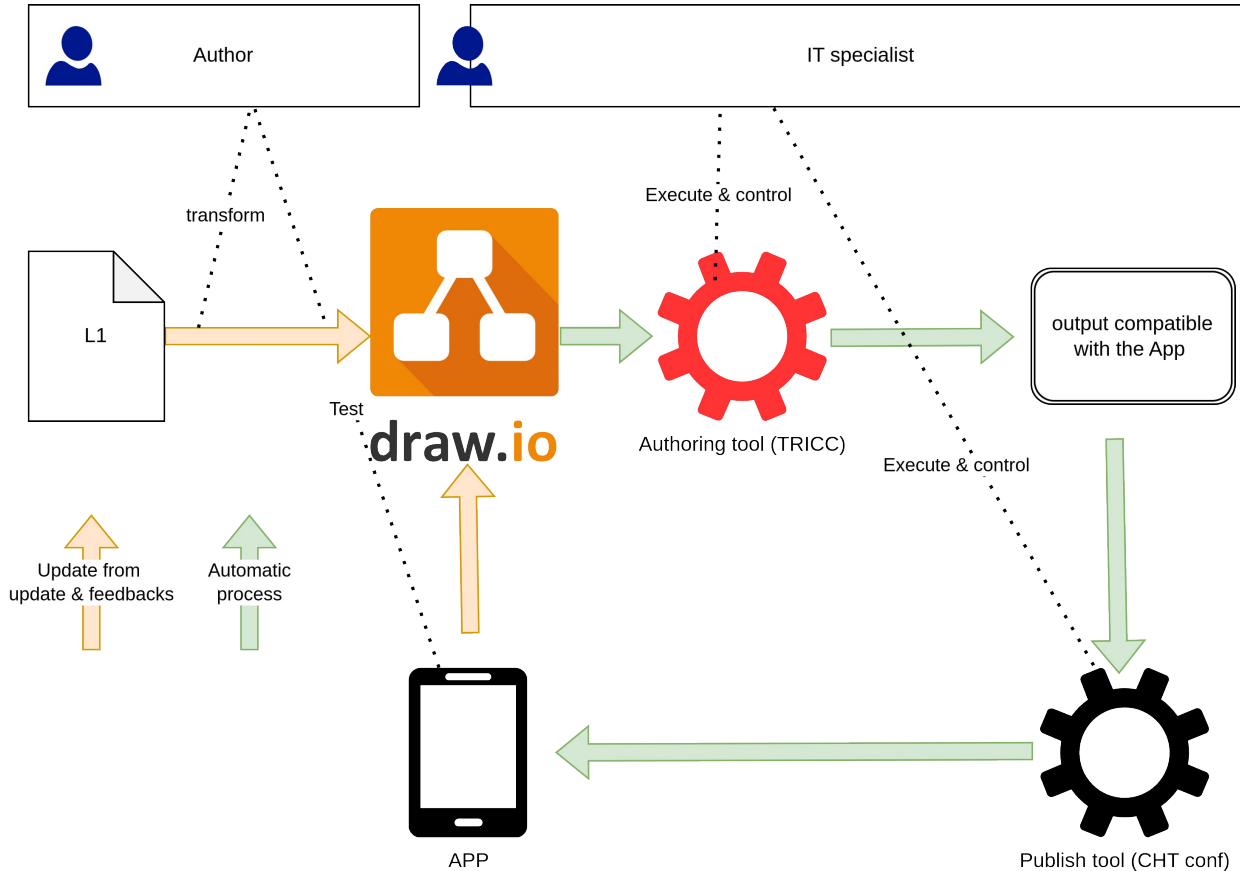
How it helps

- **Collaboration:** Decision tree diagram can be read by many with limited training
- **Intuitive Understanding:** Calculation with masked details enable precision while not making the content overwhelming at first sight
- **Treatments:** Treatment is managed as a segment with its own decision trees
- **Activity Management:** Visual authoring is centered around Activity definitions reuse
- **Sequence logic:** Visual authoring implements the capabilities we required in the past, no need to reinvent the wheel
- **When to Start:** Visual authoring is done per segment, making each part easier to understand, and allowing several segment version to run in parallel
- **Conversion:** Visual authoring comes with automated content conversion, ensuring adherence to the visual & Comutable Clinicial reference files

Visual Authoring Implementation

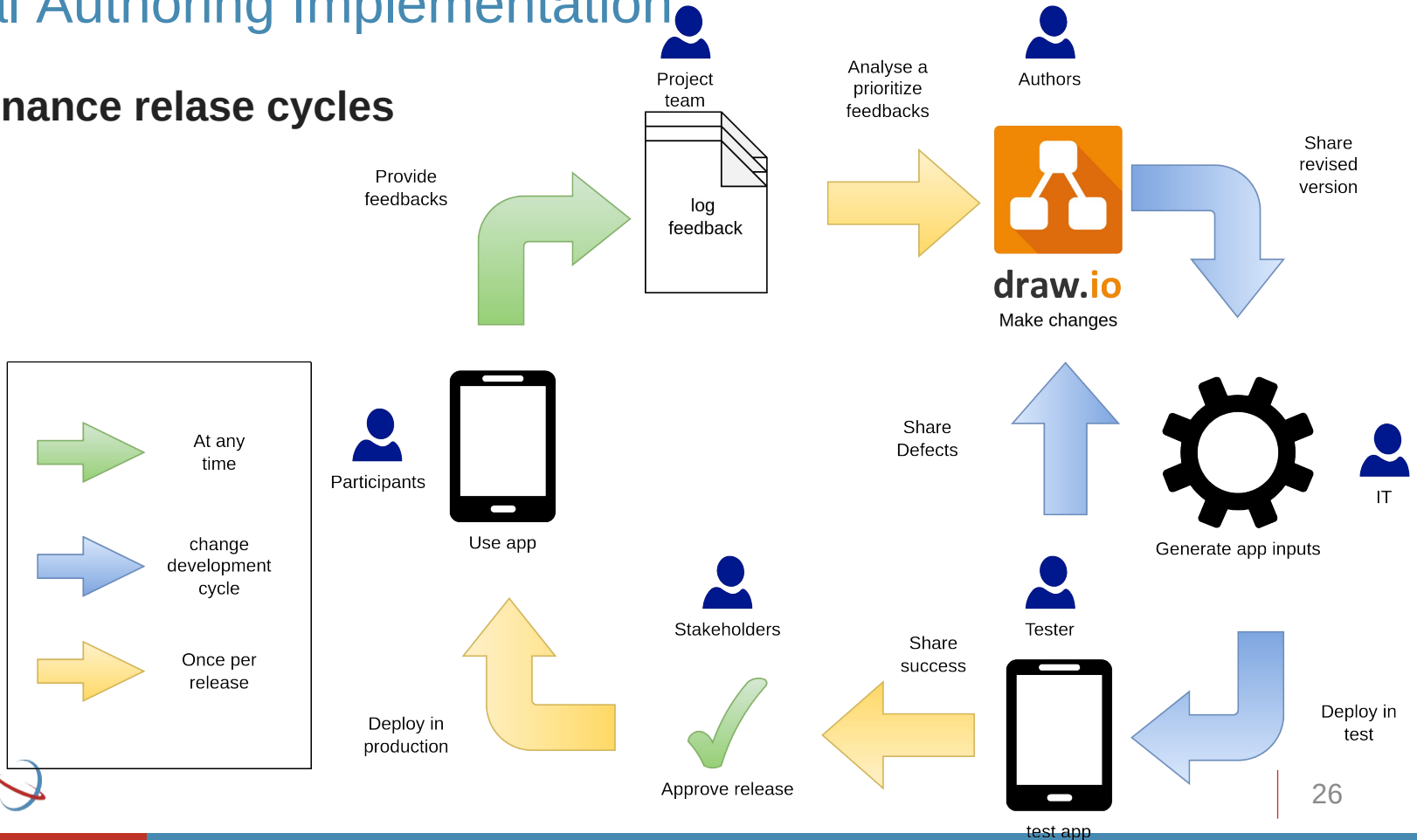
Visual Authoring Implementation

Process



Visual Authoring Implementation

Maintenance release cycles



Visual Authoring

Tooling

Drawio:

- Used to make the diagram
- Opensource - free
- <https://app.diagrams.net/>

Tricc:

- Conversion tool, takes drawio files as inputs
- Opensource - free
- <https://github.com/SwissTPH/tricc>

Visual Authoring Implementation

Recommendations

- **Keep visualisation manageable:**
 - Few seconds should be enough to get an idea of what they do
 - Put complex logic in CQL to avoid over complicating them
- **Maintain a data dictionary**
 - Use code for all the concept
 - Define properly your concept to avoid having the associated data misused
- **Small iteration** : Small update, convert, test
 - Each addition make the overall more complex, giving it more reason to have issues (undefined elements, loops, ...)
 - Small scope ease the localization of the breaking part in case of errors

Visual Authoring Implementation

Recommendations

- **Change control**
 - Latest version is shared on collaboration platform
 - At least 1 file per segment to allow // authoring without merging issues
- **Computation Troubleshooting**
 - Read carefully the log (warning and error) before jumping to conclusion:
 - Failed import, processing
 - Undefined concept
 - In the case of a possible “lock or loop”, check your modification for:
 - activity/node that leads to itself (may be with many nodes in-between)
 - Concept expected for calculation but defined in a later activity

Visual Authoring Demo

CDSS Autoring remaining challenges

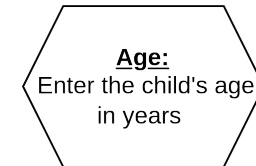
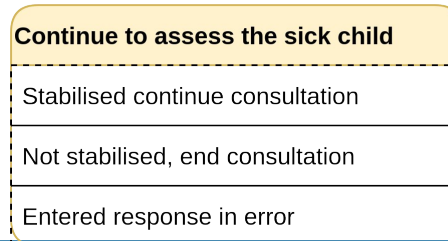
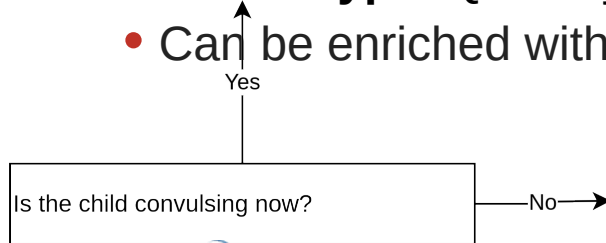
- Front end User experience need to be improved
 - More foolproof
 - Allow collaboration
- Higher linkage to codesystem and mapping capabilities should be available
- Stronger treatment data modeling to manage treatment regement and replacement drugs
- Technology Agnostic testing

Visual Authoring

Nodes / Tasks

Data capture tasks/nodes

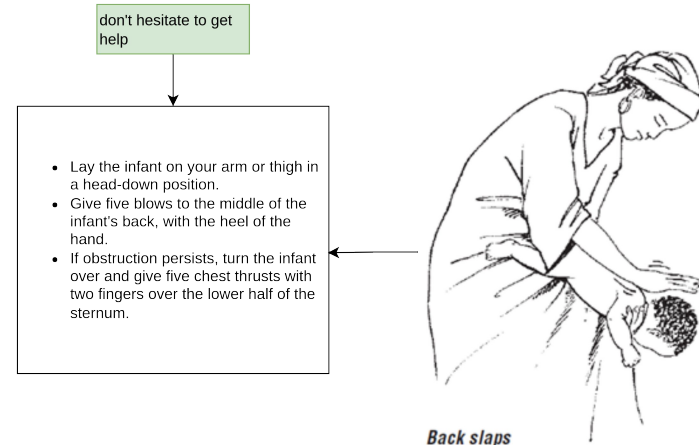
- **Purpose:** Capture inputs from the user
- **Concept:** collect data based on:
 - The concept to be collected: (Code, Type of data expected)
 - Data collections conditions (validation, error message)
 - **Supported types:** Date, Number, Text, Choice (select_one/multiple), Booleans (YesNo)
 - **Future type:** Quantity
 - Can be enriched with images, hint or help messages



Visual Authoring tasks/nodes

Message node

- **Purpose:** display information to the user
- **Concept:**
 - Can be enriched with images, hint-message or help-message
 - May integrate calculated values



Visual Authoring tasks/nodes

Calculate node

- **Purpose:** allow calculation or saving if a path was reached
- **Concept:**
 - Logic Definition: allows complex logic with CQL (calculate)
 - “Reference” attribute contains the CQL
 - Tool to help CQL writing : <https://thespark.dev/>
 - Sequence based logic:
 - Add (add)
 - Count (count)
 - Boolean (calculate)

Visual Authoring tasks/nodes

Sequence node

- **Purpose:** allow comprehensive decision trees
- **Concept:**
 - decision node (rhombus)
 - End nodes (end, activity_end)
 - Conditional edges
 - Activity GoTo (goto)
 - Start nodes (start, activity_start)
 - Bridge (bridge)
 - Wait (wait)
 - Inputs (inputs)

Visual Authoring tasks/nodes

Sequence node : Inputs

- **Purpose:** define external element that will be populated
- **Concept:**
 - Calculate might reference data not captured by the authored content but populated before the encounter start
 - Ex: Malaria risk

Visual Authoring tasks/nodes

Sequence node : Rhombus

- **Purpose:** create a fork in the decision logic
- **Concept:**
 - Conditional edges works like rhombus, label containing the CQL
 - “Reference” attribute contains the CQL or the node that must be `true`
 - “name” is not important and can be left to `r_`
 - Out edge must be on of:
 - yes: path followed if the CQL is evaluated to `true`
 - No: path followed if the CQL is evaluated to NOT `true` (null and false)
 - Continue / Follow: rhombus serves as a bridge (reference not used)

Visual Authoring tasks/nodes

Sequence node : Activity GoTo

- **Purpose:** reference an activity to be performed
- **Concept:**
 - Place holder that will be replaced by the actual activity
 - “link”: rely on the drawio “link to other tab“ data structure
- **Key advantages:**
 - Enable modular approach

Visual Authoring tasks/nodes

Sequence node : Start

- **Purpose:** define where a diagram starts
- **Concept:**
 - start: start of a process
 - “process” define which process it starts
 - activity_start: start of an activity
 - Both start may have relevance that will condition the complete activity/process

Visual Authoring tasks/nodes

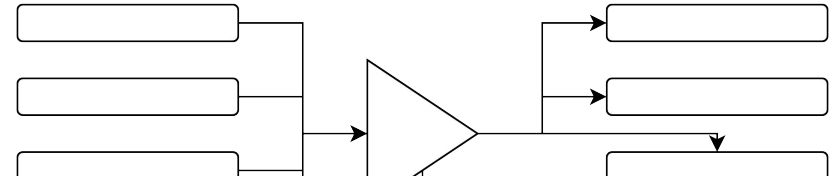
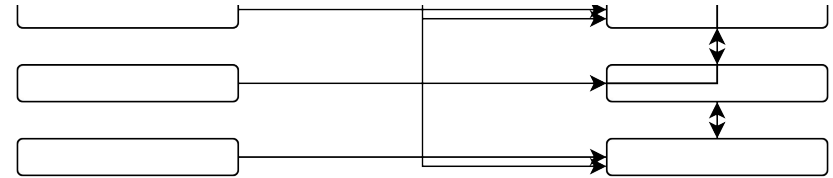
Sequence node : wait

- **Purpose:** ensure following node is processed after waited nodes
- **Concept:**
 - “reference” containing the waited node

Visual Authoring tasks/nodes

Sequence node : Bridge

- **Purpose:** create point where several path converges
- **Key advantage:**
 - Help readability when many path should go in list of target
 - 4 path in in 4 targets means 16 edges without bridge, 8 with bridge



Visual Authoring tasks/nodes

Sequence node : link to

- **Purpose:** replace a edge to increase visibility
- **concepts:**
 - Reference should be shared between the link in and to
 - Not used TBC if still working

Visual Authoring tasks/nodes

Sequence node: ends

- **Purpose:** terminate a part or the full decision tree
- **Concept:**
 - **Activity end:** that once reached by all its path make the activity as done
 - **Pause/breakpoints:** to pause an encounter in case external activities (tests ...) are required (“process”=pause)
 - **End:** ends the encounter completely
 - **End to:** break the encounter until the named segment (e.g. For accelerated pathway, put in “process” the process you and to go to)