ChatGPT

Workflows for Youbooks-Based Book Creation and Maintenance

Project: *Founding an Ecosystem for Wealth Transition*
Objective: Create a full-length interdisciplinary book and derived summaries, compatible with Youbooks (AI book generator) and multiple output formats, plus a companion workbook.

Inaugural Workflow: From Concept to First Edition

This workflow details how to go from an initial idea to a publishable first edition, leveraging Youbooks for AI-assisted drafting and external tools for figures, cross-references, and multi-format outputs.

Stage 1: Planning and Outline Development

- **Define Scope & Audience:** Clearly identify the book's goals, target readers (e.g. wealth transition professionals, event sponsors), and tone. In this case, the aim is to articulate the *why* and *how* of founding an ecosystem of professionals for family wealth transition, in a professional yet accessible tone [1] .
- **Outline Structure:** Develop a detailed Table of Contents before writing. The user's skeleton outline (Parts I–IV with 21 chapters plus appendices [2]) serves as a roadmap. Refine this outline to ensure logical flow (e.g. Part I sets context, Part II profiles disciplines, Part III describes the "Founding" event, Part IV looks at sustaining the ecosystem). Each chapter's purpose is defined in brief bullet points (as in the prompt) to guide content creation [3] [4] .
- **Research & Source Gathering:** Compile any source material to inform the text (e.g. case studies like Rockefeller, interdisciplinary best practices). If using Youbooks, gather reference documents (PDFs, articles) that can be uploaded as sources to influence the AI generation (Youbooks supports PDFs/EPUBs as input) [5] . Also list authoritative works to cite in the bibliography across law, finance, psychology, etc. [6] .
- **Tooling:** Use an outlining tool or writing software (such as **Scrivener** or a markdown editor) to capture the structure. Scrivener excels at organizing a manuscript into parts and chapters, allowing you to manage chunks of text easily [7] . If using Scrivener, create folders for each Part and documents for each chapter as per the outline. This modular setup will later aid in targeted updates and compilation.

Stage 2: AI-Assisted Drafting with Youbooks

- **Initial Manuscript Generation:** Leverage **Youbooks AI** to generate a draft for each chapter using the outline and descriptions. Youbooks can take a detailed outline or description of each section and produce well-structured content in a matter of hours [8] [9] . It employs a multi-step AI workflow (over 1,000 refinement steps) with multiple models (e.g. GPT-4, Claude) to research, structure, and write the text [10]. This results in a coherent non-fiction manuscript covering all specified chapters and topics [11].

- **Input to Youbooks:** Provide the Title, audience, and outline (as prepared in Stage 1) to Youbooks. Specify any *tone/style* requirements (the prompt calls for professional yet accessible language, avoiding undefined jargon [1]) so the AI's output aligns with your voice. If Youbooks allows it, choose a mode where your sources "Strongly influence" the content [12] to ensure accuracy in case studies and definitions (e.g. definitions of "Founding" and "Wealth" from the prompt [13] are correctly included).
- **Text-Only Draft:** Note that Youbooks will generate **text only** – it does not create figures or images. The output can be downloaded in editable formats such as **Markdown, DOCX, or PDF** [14] [15]. Markdown is ideal for our purposes because it preserves structure (headings, lists, etc.) and is easily processed by external tools. Save the AI-generated book manuscript as a Markdown (.md) file or import the DOCX into Scrivener for further work.

Stage 3: Human Revision and Expansion

- **Content Review:** Treat the AI draft as a starting point. The author must now revise each chapter for accuracy, clarity, and depth. Check facts and fill any gaps: e.g. verify the historical case studies (did the AI accurately recount Rockefeller's wealth transition example?), ensure discipline-specific chapters contain nuanced, up-to-date insights (tax law changes, psychological perspectives, etc.). Remove any AI hallucinations or irrelevant tangents.
- **Enhance and Customize:** Infuse the manuscript with the author's expertise and voice. Add anecdotes, real-world examples, or quotes from experts to enrich the content. Ensure each chapter addresses the points in the outline. For example, if Chapter 4 (Estate Planning Attorneys) lacks a "pitfalls when working in isolation" section (per outline) [16], write a paragraph on that. Maintain consistency in terminology (possibly add key terms that emerged to the Glossary draft).
- **Style and Tone Editing:** Edit for a cohesive tone across chapters. The audience is interdisciplinary, so keep language accessible and define technical terms. Use an editing tool like **Grammarly** or **ProWritingAid** to catch grammatical issues and ensure readability. At this stage, an LLM can help refine prose: e.g. ask ChatGPT to suggest smoother phrasing for complex passages or to ensure the introduction and conclusion tie together well. The author remains in control, accepting changes that improve clarity and rejecting those that alter meaning.
- **Tools:** Continue working in Scrivener or a markdown editor. If using Scrivener, you can use the split-screen view to see chapter content alongside research notes. If using markdown files per chapter, a code editor or Markdown editor (VSCode, Typora) can be used for revision. Version control can begin here: consider initiating a **Git repository** for the project. This will track all changes going forward and facilitate branching for future versions (commit the initial AI draft, then commit after each revision pass to have a history). Git is well-suited for writers because it keeps everything in one folder and allows easy version snapshots without dozens of separate files [17].

Stage 4: Figure Planning and Development

- **Identify Figure Needs:** Review the text for places where a figure (diagram, chart, or illustration) would enhance understanding. For instance, in an interdisciplinary concept like this, you might include a **diagram of the ecosystem** (showing how estate lawyers, tax advisors, therapists, etc. connect) or a flowchart of the "Founding" event stages (Part III). Mark in the manuscript where these should appear (e.g., "

$$Figure : Venndiagramofoverlappingdisciplineshere$$

”). Each figure should have a descriptive caption so that in a text-only environment (like Youbooks platform or EPUB without images) the reader still gets the information. For example: *“Figure 3.1: Overlapping Roles of Disciplines in a Wealth Transition Ecosystem (a Venn diagram illustrating shared responsibilities).”*

- **User-Supplied vs AI-Generated Figures:** For each needed figure, decide the creation method:
- *User-Supplied:* If you already have charts or graphics (perhaps slides from presentations on wealth transition), adapt those. Ensure they are high resolution and clear in grayscale (if printed).
- *AI-Generated:* For conceptual or illustrative images you don’t have, use AI image generation. Text-to-image models (like DALL·E or Midjourney) can create illustrative figures from prompts. For example, prompt an AI for “a flowchart showing stages of a multi-day professional workshop event” or “an infographic of disciplines working together.” You may need to iterate and then possibly edit the AI image in an image editor for polish. **Tip:** If the figure is a diagram with text (labels), it’s often better to create it manually with a drawing tool so the text is crisp.
- *Manual Design:* Use diagramming tools (e.g. **diagrams.net/draw.io**, **Microsoft Visio**, or **Lucidchart**) for process flows or relationship maps. These allow precise control and labeling. Or use **Inkscape/ Adobe Illustrator** for more freeform illustrations.
- **Figure Creation:** Proceed to create each figure outside the manuscript: design or generate the image, then save it with a clear filename (e.g., `Fig3.1-EcosystemDiagram.png`). Maintain an **“images” folder** in your project directory to store all figure files. Keep track of figure numbering as per chapters.
- **Caption and Alt-Text:** Write a caption for each figure and an alt-text (a short textual description) for accessibility. The caption goes into the manuscript (it will appear in the List of Figures and near the image in print), and alt-text can be used for the web version or EPUB for screen readers. For example, for the Venn diagram: *“Figure 3.1: A Venn diagram illustrating overlapping expertise among disciplines (e.g., both attorneys and financial planners deal with tax implications).”* Alt-text might be: *“Venn diagram shows overlap between estate planning, tax advising, and asset management.”*
- **Manuscript Integration (Text-Only):** In the text, reference the figures by their number (which we will manage dynamically later). For now, you can insert a placeholder image reference in Markdown:

```
![Overlapping Roles of Disciplines](images/Fig_EcosystemVenn.png)
{#fig:ecosystem}
*Figure: Overlapping roles of various disciplines in a wealth transition
ecosystem.*
```

The above Markdown (with a pandoc cross-reference ID) ensures the image will be included in outputs and labeled. If working in Scrivener, you might not insert the actual image yet, but put a marker like “<<Insert Fig3.1 about ecosystem here>>”. The key is that the manuscript text describes the figure’s content so a Youbooks reader isn’t left guessing.

Stage 5: Cross-Reference and Citation Management

- **Automate Numbering:** Use tools to automatically handle numbering of chapters, figures, tables, and cross-references. If writing in Markdown, adopt **Pandoc with cross-reference filters** (like `pandoc-crossref`). For example, using pandoc-crossref, the figure syntax above will auto-number the figure and allow in-text references like “As shown in @fig:ecosystem, the disciplines overlap…” to be resolved to “Figure 3.1” [18]. This approach works for multiple output formats (HTML, PDF, etc.) [19]. In LaTeX, you would use `\label{fig:ecosystem}` and `\ref{fig:ecosystem}` for the same effect. **Scrivener note:** Scrivener’s compile can preserve MultiMarkdown tags or placeholder tags for images; if using Pandoc via Scrivener, you can still rely on the Markdown syntax and have Pandoc do the heavy lifting after compile.

- **Section and Appendix References:** If the text says “see Chapter 17 for more on Founding stages,” ensure each chapter or section has an identifier so it can be referenced. In Markdown/Pandoc, you can add `{#sec:founding-stages}` after a chapter title, and refer to it as “Section 17.2” with pandoc-crossref. In LaTeX, use `\label{chap:FoundingStages}` after the chapter title and reference with `\ref`. This guarantees these references stay updated if chapter numbers change.
- **Citations and Bibliography:** Incorporate citations for external sources. Since the book spans multiple domains, compile a **BibTeX** `.bib` file (or use another citation format supported by Pandoc) with references for all sources (books, articles, case studies). As you revise the text, insert citations like `[@Doe2020]` (Pandoc/CSL syntax) or use Scrivener’s placeholder tags for references, which can later be replaced by formatted citations. Choose a citation style (APA, Chicago, etc.) that fits the book’s tone. Using Pandoc’s `--citeproc` (or the built-in citeproc in newer Pandoc) will automatically generate a **Bibliography** section from these citations [20]. Ensure the Bibliography gets appended as an appendix.
- **Glossary Creation:** Build the **Glossary** appendix in parallel. As you edit, keep a list of key terms (like *Founding, Cadre, Wealth Transition, Philanthropic Asset,* etc.) and their definitions. One strategy: create a separate Markdown file (e.g. `glossary.md`) where you add entries as you encounter new terms in the text. Alternatively, mark glossary terms in the text (e.g. in Scrivener, apply a character style “GlossaryTerm” each time a term appears) so you can find them later. Ultimately, alphabetize and format the glossary in the appendix. This can be done manually or semi-automatically (e.g. export the list of terms marked and sort them).
- **Index Preparation:** Decide on how to generate the **Index**. For the inaugural edition, you might manually select important names, concepts, and sub-topics for indexing (because a purely automatic index could include too much trivial detail). Mark index entries in text as you revise: in LaTeX, use `\index{Term}` for each occurrence; in Word or Scrivener, you might use an Index marking tool or placeholder. Pandoc doesn’t index by default, but a Lua filter or a subsequent LaTeX run with `makeindex` can produce one. A reasonable approach: after the manuscript is final, run a quick scan for names (e.g., “Rockefeller”) or key concepts (e.g., “governance”) and ensure they are marked for the index. The index will be generated during compilation (see Stage 7).
- **Cross-Referencing Tools:** If using **Scrivener**, you may rely on Scrivener’s placeholders (e.g. linking to a section by name) – however, these might not cover figures and tables well. It can be simpler to compile to Markdown and run Pandoc with crossref to handle everything. On the other hand, **Microsoft Word** users could wait until the content is in Word, then use Word’s reference features (captions and cross-references for figures, automatic table of figures, indexing tool, etc.). But that is more manual and less reproducible for future updates, so leveraging Pandoc/LaTeX automation is preferable for a complex book.

Stage 6: Versioning and Branching Setup (Preserve First Edition)

- **Establish Version Control:** Now that the manuscript text and supporting files (images, .bib, etc.) are in a stable state, set up a robust versioning system. Using **Git** is highly recommended to manage revisions. All project files (Markdown chapters, image files, bibliography, and supplementary material like the summary documents and workbook) should reside in a single repository folder [17].

Commit the “First Edition” final draft as a baseline. Tag it as `v1.0` (for example). This way, after publication, you can always retrieve the exact state of the first edition, and later start making changes on a new branch for edition 2. Git’s branching feature allows you to create a branch (e.g., `second_edition`) where you implement updates while keeping the original intact [21]. (If not using Git, at least duplicate the project folder and label it clearly for new versions, though this is less efficient.)

- **Parallel Outputs Branching:** Consider maintaining separate branches or subfolders for different output formats if they diverge (for instance, a branch where you make layout tweaks for print that you don’t need in the e-book version). Generally, this isn’t necessary if using one source text and Pandoc, since you can generate all formats from the same source. Keep the *prospectus/summaries* documents in the same repo; these can be updated alongside the book content and you can tag versions of them too.

Stage 7: Multi-Format Compilation of the Book

Now generate the primary deliverable – the full book – in various publishable formats. Automation is key here to avoid manual reformatting for each version.

- **Choose a Compilation Toolchain:** We will use **Pandoc** for its ability to convert Markdown to PDF, Word, HTML, etc., while applying templates and filters for features like cross-references and citations. Pandoc can take the markdown manuscript (with embedded figure links and citations) and produce a print-ready PDF via LaTeX, as well as other formats [7] [22]. Alternatively, you could use Scrivener’s compile features to generate a DOCX or LaTeX file and then refine with Pandoc/LaTeX.
- **LaTeX/PDF Production:** For a high-quality PDF (suitable for print or professional distribution), Pandoc can use a LaTeX engine. For example:

```
pandoc book.md --toc -N \
       --filter pandoc-crossref --citeproc \
       -V geometry:margin=1in -V fontsize:11pt \
       -V documentclass:book \
       -V papersize:letter \
       -o "Founding_Ecosystem_Book.pdf"
```

This command would take `book.md` (which is the merged Markdown of all chapters and appendices) and produce a PDF with a table of contents (`--toc`), numbered sections (`-N`), cross-references resolved, citations processed into a bibliography, using a book class template with certain margins and font size. You can customize Pandoc’s template or use a custom LaTeX template to include front matter (e.g., a title page) and back matter (index). For the **Index**, if you inserted `\index{}` tags in the LaTeX output, you need to run `makeindex` on the intermediate `.tex` file, then compile again. Pandoc’s workflow can’t do that in one step, so it might be easier to generate a LaTeX file (`.tex`), run LaTeX manually:

- `pandoc book.md ... -o book.tex`
- Run `pdflatex book.tex`, then `makeindex book.idx`, then `pdflatex book.tex` twice more. This yields a PDF with an index.

You can script these steps (e.g., a Bash script) so it’s one command to regenerate the book. **Tool:** A simple Bash script or Makefile can automate multi-step builds (Pandoc -> LaTeX -> PDF + index, etc.).

- **DOCX (Word) Output:** Use Pandoc to generate a Word document as well, for stakeholders who may prefer reviewing in Word or for future editing:

  ```
  pandoc book.md --filter pandoc-crossref --citeproc \
         -o "Founding_Ecosystem_Book.docx"
  ```

  Pandoc will convert headings, lists, etc. to a structured Word file. You can supply a reference DOCX (with predefined styles) to control the look (e.g., if you have a Word template with specific font/ heading styles). The cross-references will appear as hyperlinked text (e.g., "Figure 3.1") and the bibliography and ToC can be embedded as well. A quick check in Word may be needed to ensure the List of Figures and Index came through (Word might not automatically regenerate them, so you might generate those sections as static text via Pandoc or update fields in Word).
- **EPUB/HTML:** Although not explicitly requested, you might also produce an **EPUB** e-book or **HTML** for web reading, since Pandoc makes it easy. This could be future-proofing. For example:

  ```
  pandoc book.md -o FoundingEcosystem.epub
  ```

  yields an EPUB that could be distributed for e-readers. The text-only nature of Youbooks content means it's well-suited for an EPUB (images will be embedded).
- **Quality Checks:** After generating, review the PDF and DOCX. Ensure the **Table of Contents** is present and correct, **List of Figures** is generated (in LaTeX, including `\listoffigures` in the template or using a Pandoc filter for it [23] can auto-create this list), the **Glossary** and **Bibliography** appear formatted, and the **Index** is at the end. Make sure page numbering, header/footer (if any), and overall layout are professional. Adjust Pandoc template variables or LaTeX preamble as needed (for instance, add `\usepackage{caption}` to adjust figure caption style [24]). This might require a couple of iterations.
- **Scrivener Alternative:** If working through Scrivener, an alternate path is to compile to multiple formats via a tool like **Scrivomatic** (which integrates Scrivener with Pandoc). Scrivener can use its MultiMarkdown compile to trigger Pandoc conversions automatically [25] [26] . For example, you could set up a compile format in Scrivener that produces PDF, DOCX, and HTML in one go [22] . In either case, the goal is a *single source* (the manuscript text) feeding *multiple outputs*. This ensures consistency across formats and simplifies maintenance.

Stage 8: Creating the Prospectuses and Summaries

With the full book content finalized and generated, derive the shorter formats required for different audiences. The project specifies three summary formats plus an executive summary, which we interpret as:

- **Full-length Prospectus (5–10 pages):** Essentially a detailed summary or condensed version of the book, touching on each chapter's key points.
- **One-Page Prospectus:** A very brief overview for busy readers, highlighting the core message and hook of the book in one page.
- **Executive Summary:** A mid-length summary (perhaps ~2 pages) focusing on strategic insights and high-level conclusions, suitable for executives or sponsors.
  (The terms sometimes overlap; we'll treat the "executive summary" as distinct from the one-page, likely a 2-page brief that is more narrative than the bullet-point style 5–10 page prospectus.)

Workflow to produce these:

- **Summarize Each Chapter:** Use a combination of author skill and AI assistance to extract the essence of each chapter. One approach is to have the author write a few sentences per chapter capturing the main idea and any crucial sub-points. Alternatively, use an LLM (like GPT-4) to assist: provide each chapter text (if short enough) or a detailed outline of it to the AI and ask for a concise summary. For consistency, maintain a neutral informative tone in these summaries (since they will be compiled together). For example, Chapter 1's summary might say: *"Chapter 1 introduces the rationale for founding an ecosystem, citing generational wealth transition challenges and the failures of siloed advice. It sets a vision for a 'Founding' event as a novel solution."* Do this for all chapters.
- **Compile the 5–10 Page Prospectus:** Organize the chapter summaries into a coherent document. It could mirror the book's structure, possibly grouping chapters by part or theme for flow. Ensure the prospectus isn't just a list of disjointed summaries – include a brief introduction and conclusion to frame it as a standalone overview. The introduction (one paragraph) should describe the overall purpose of the book and the importance of the ecosystem approach, to engage sponsors. Then each chapter summary follows (you can use bullet points or short paragraphs for each). Keep the whole thing within 5–10 pages (perhaps ~2,500–3,000 words). This might require trimming some chapter summaries to one or two sentences to fit length.
- **Write the One-Page Prospectus:** This is a highly distilled sales pitch for the book. Focus on the *big picture and value proposition*: Why does the ecosystem matter? What pressing problem does the Founding solve? Include perhaps a sentence or two about each Part of the book rather than each chapter. For example: a paragraph on the challenges of wealth transition today, a paragraph on the solution (the interdisciplinary cadre and Founding event), and a paragraph on outcomes (resilience, better legacy outcomes, etc.). Aim for a single page (~300-500 words). Tools like GPT-4 can be tasked with *"Summarize the entire book in one page, targeting an audience of potential sponsors, and make it engaging"*. The AI output can then be edited for punchiness. Ensure it has a compelling opening line and a clear call-to-action or inspirational closing that would make someone want to support the initiative.
- **Prepare the Executive Summary (~2 pages):** If this format is needed distinct from the above, consider it a slightly expanded one-pager: ~2 pages (600-1000 words) that cover key findings or recommendations. Unlike the prospectus which might stick to describing the content, an executive summary might focus on outcomes and actionable insights. For example, highlight 3–5 key takeaways or proposals from the book: *families in wealth transition need a team approach; regular interdisciplinary retreats ("Foundings") can preempt conflicts; investment in such ecosystems yields long-term family and societal benefits*, etc. You might bullet these and then provide a brief supporting explanation for each. Since this overlaps with the 5–10 page summary's content, you can indeed derive it by condensing that intermediate summary further, or by taking the one-pager and elaborating slightly on each point.
- **Use LLM for Consistency:** After drafting the summaries, you can use an LLM to review them for consistency and clarity. For instance, *"Please proofread and ensure the tone of this 5-page summary is professional and that it logically flows from one chapter to the next."* The AI can also help reduce redundancy. Always do a final human edit to ensure no important nuance was lost.
- **Formatting the Prospectuses:** Treat each of these as separate documents. Use Markdown for easy conversion. You might have `prospectus_full.md`, `prospectus_onepage.md`, and `exec_summary.md`. Each can have its own small table of contents if needed (though one-page won't). They should be stand-alone, so include the book title and author at top for context and maybe a note like "Summary of *Founding an Ecosystem for Wealth Transition* by [Author]."

- **Conversion to Outputs:** Use Pandoc (or Scrivener) to convert these summaries to **PDF, Word, and HTML** as needed, similar to the book. For example:

  ```
  pandoc prospectus_full.md -o FoundingEcosystem_Prospectus.pdf
  pandoc prospectus_onepage.md -o FoundingEcosystem_OnePage.pdf
  ```

  and so on, as well as `.docx` versions if desired. The styling can be simple. Since these are marketing materials, you might later format the PDF versions with more design (like adding logo or header). But the core content is handled via the same pipeline for consistency.

- **WXR Package for WordPress:** To satisfy the requirement of having all three prospectus formats and the executive summary importable to a WordPress site, create a **WXR (WordPress Extended RSS) file** containing them. WordPress uses WXR (an XML format based on RSS) to import posts/pages [27]. The strategy: convert each summary to HTML (Pandoc `-t html` or even simply rename the .md to .html if it's basic). Then wrap each in an `<item>` entry within a WXR XML file, with appropriate titles and possibly a category (you might categorize them as "Prospectus" pages). Tools for this step: you can manually write a small XML (using a template from a WP export) or use a script. If not comfortable hand-coding XML, an easier method is: *Create a temporary WordPress site*, copy-paste the summaries as pages, use WP's **Tools → Export** to get a WXR file of just those pages. That WXR can then be imported to the target site. This ensures formatting (like bullet points or bold text) is preserved. In any case, the deliverable is a single `.xml` file that contains four items (Full prospectus, One-page, Executive summary, maybe also the intermediate if that's counted separately) as WordPress entries.

Deliverables at this point: You now have: (1) the Full Book in PDF/DOCX (and possibly EPUB), (2) a 5–10 page prospectus (PDF/DOCX/HTML), (3) a one-page prospectus (PDF/DOCX/HTML), (4) a ~2-page executive summary (PDF/DOCX/HTML), and (5) a WXR file containing the summaries for web import. The content across these is consistent (all derived from the master manuscript). The next step is the workbook.

<table> <tr><th>Output Deliverable</th><th>Content Source & Creation</th><th>Generation Tools/ Format</th></tr> <tr><td>**Full-Length Book** (1st Edition)</td><td>Complete manuscript (all chapters & appendices) with integrated figure references and citations.</td><td>Markdown ➔ Pandoc➔ PDF (via LaTeX template) for print; Pandoc➔ DOCX for editing; Pandoc➔ EPUB/HTML for digital. Cross-references and bibliography automated [18] [22].</td></tr> <tr><td>**Full Prospectus** (5–10 pages)</td><td>Consolidated chapter summaries (approx. one paragraph per chapter) plus intro/conclusion.</td><td>Drafted with LLM assistance; Markdown ➔ Pandoc➔ PDF/DOCX for distribution; also converted to HTML for web.</td></tr> <tr><td>**One-Page Prospectus**</td><td>High-level overview of the book's key message and scope, in a single page.</td><td>Drafted with LLM assistance (from full summary); Markdown ➔ Pandoc➔ PDF/DOCX; also to HTML.</td></tr> <tr><td>**Executive Summary** (~2 pages)</td><td>Top-line insights and recommendations, focusing on big-picture outcomes.</td><td>Authored/LLM hybrid drafting; Markdown ➔ Pandoc➔ PDF/DOCX; also to HTML.</td></tr> <tr><td>**WordPress Import (WXR)**</td><td>All the above three summaries and exec summary in one file as separate items.</td><td>Content in HTML wrapped in WXR XML structure [27]; created via script or WP Export tool. Importable to WordPress to create pages/posts for each summary format.</td></tr> <tr><td>**Printable Fillable Workbook**</td><td>Supplementary PDF workbook with exercises, prompts, and forms corresponding to book chapters.</td><td>Authored in Word/

LaTeX with form fields (text fields, checkboxes); output to PDF. Form fields added via Word Developer tools or Adobe Acrobat (for fillable functionality).</td></tr> </table>

Stage 9: Companion Workbook Development

Alongside the book, create a **printable, fillable PDF workbook** that readers (or event participants) can use to apply concepts. This workbook should complement the book chapter by chapter, offering questions, checklists, or planning templates.

- **Design the Workbook Structure:** Mirror the book's structure to some extent. For example, if Part II of the book introduces disciplines, the workbook could have a section where the reader identifies which of those disciplines they have in their network or which they need to engage. For Part III (The Founding event), the workbook might have a planning template for organizing a Founding event in the reader's context. Outline the workbook sections corresponding to key parts of the book.
- **Draft Content for Exercises:** For each chapter or major section, create exercises or reflective prompts. Some examples:
- After chapters on disciplines (estate planners, therapists, etc.), an exercise could be: *"List the professionals in each domain that your family or firm currently consults. Identify gaps in expertise and brainstorm how to fill them."*
- After the "Stages of a Founding" chapter, include a checklist for event planning or a timeline template that the reader can fill with their own dates and tasks.
- In sections about values or legacy (if covered in Part I or IV), include writing prompts for the reader to articulate their family mission or philanthropic goals.
- Glossary terms could be turned into a matching exercise or a quiz, but since this is a professional book, likely the workbook remains a serious planning document rather than a quiz.
- **Ensure Fillable Fields:** For each prompt or question, leave adequate blank space or an actual form field for the user's response. E.g., lines for writing or multi-line text boxes for longer answers. If asking yes/no or multi-choice questions, provide checkboxes or radio buttons. Label these clearly. The workbook is essentially a form, so designing it in a tool that supports form fields is ideal.
- **Tool Options for Workbook Creation:**
- *Microsoft Word:* You can create a form in Word using the Developer tools. Insert text form fields (rich text content controls) for long answers, and check box form controls for checklists. Format the document with appropriate section headings, instructions, and lines or boxes where needed. Once the layout is ready, save as PDF – Word's form controls can be made to remain fillable in the PDF. (On export, ensure "Create bookmarks" for headings is enabled if you want a PDF navigation, and that interactive form fields are preserved – in recent Word versions, basic fields do carry over to PDF).
- *LaTeX:* If you prefer, LaTeX with the **hyperref** package can create PDF forms. For example, `\TextField{Name:}` creates a text field form element [28]. This requires some scripting, but one could use a LaTeX template to layout questions and use `\TextField`, `\CheckBox` etc. This yields a very polished PDF, but the learning curve is higher. There are tutorials on using LaTeX to create forms [29]. If you already have a Pandoc workflow, you could even write the workbook content in Markdown and use a LaTeX template that inserts form fields at the right spots.
- *Adobe Acrobat:* Another approach is to design a static PDF (maybe exported from Word or InDesign) and then use Acrobat's **Prepare Form** tool to draw form fields on top of the blanks. This is often straightforward: Acrobat will detect underlines as potential text fields or empty checkboxes as check fields, and you can tweak from there. The result is a fillable PDF.

- *InDesign or Publisher:* For high-end design, a tool like Adobe InDesign can layout a nice-looking workbook with graphical elements, then export to PDF and, if needed, one can still add form fields in Acrobat. However, using InDesign is likely overkill unless design aesthetics are crucial.
- **Populate and Format:** Write out all questions and instructions in your chosen tool. Keep paragraphs short and use bullet lists or numbered steps for multi-part exercises (keeping with clear formatting guidelines). For instance, use numbered lists for step-by-step planning tasks, and bullet lists for collections of ideas to brainstorm. Clearly separate sections by headings (so a reader can easily find "Section 3: Historical Lessons – Reflection Exercise"). Ensure consistent font and style with the main book (e.g., use the same font family for a professional look, maybe a serif font for body text).
- **Review and Testing:** Before finalizing, test the workbook's usability:
- Print a copy on paper to see if the layout is comfortable for handwriting (are lines adequately spaced? is there enough room for answers?).
- Test the digital PDF on Adobe Reader or other PDF viewers: can you click into each field and type? Are the checkboxes working? Is the tab order logical (pressing Tab jumps to the next field in the expected sequence)?
- Check that any instructions in the workbook reference the book correctly (e.g., "Refer back to Chapter 5 if you need examples" — those chapter numbers must match the final book).
- **Finalize Workbook PDF:** Once satisfied, generate the final **workbook.pdf**. This is a standalone product to be provided alongside the book. It should have a cover or title page as well (e.g., "*Founding an Ecosystem for Wealth Transition – Workbook*") and possibly a brief note on how to use it (for example, "This workbook is designed to accompany the book. It can be filled digitally or printed.").

Stage 10: Final Quality Assurance and Publishing

- **Proofreading:** Do one more comprehensive proofread of all outputs – the full book, summaries, and workbook. Check for any inconsistencies or typos. Given the iterative nature of AI content, ensure no placeholder text or artifacts from generation remain. Also verify that all figure callouts in text have corresponding actual figures in the PDF, and all cross-references (like "see Figure 7.2" or "discussed in Chapter 12") are resolved correctly.
- **Consistency Check:** Ensure the terminology and data are consistent between the book and workbook. For instance, if the book mentions a 5-step process and the workbook has a checklist for it, make sure it's indeed 5 steps and labeled the same. If any changes were made late in editing (e.g., renaming a concept), propagate those to the workbook and summaries.
- **Finalize Metadata:** If using LaTeX/Pandoc, ensure the title, author name, and perhaps a subtitle are correctly set in the PDF metadata and title page. Similarly, for the WordPress content, ensure titles of pages are set appropriately when importing.
- **Publishing/Export:** Now that everything is ready:
- **Youbooks Platform:** If the final text is to be available on Youbooks, you might re-upload or update the text there (without figures, since it's text-only) for consistency, but likely the main distribution will be via your own outputs now. (Youbooks was a tool to generate content, not necessarily the publishing platform, though it can output EPUB/PDF as noted).
- **Print and eBook Publishing:** Use the PDF for print-on-demand or printing press. For example, upload to Amazon KDP or IngramSpark if self-publishing, ensuring it meets their criteria (embedding fonts via LaTeX or Word ensures a proper PDF/X usually). For e-book, the EPUB or the Word file (converted to EPUB) can be used.
- **Website/WordPress:** Import the WXR file on the WordPress site (via **Tools → Import**). This will create pages or posts for the prospectus and summaries. Verify on the site that formatting looks

good (lists, bold text, etc., should appear as intended). If images were to be embedded (unlikely in the summaries, as they are text), ensure they came through or add them manually to the media library.

- **Distribution of Workbook:** Make the workbook PDF available (e.g., on the website as a download link). If it's fillable, instruct users to download and open with a PDF reader (some browser PDF viewers might not save filled data, whereas Adobe Reader will). If printing for an event, ensure high-quality prints.
- **Appendices Completion:** Double-check that appendices (Glossary, Bibliography, List of Figures, Index) are complete and included in the book. For example, the **Glossary** should list all key terms introduced (perhaps the ones given in the prompt plus any new ones). The **Bibliography** should be generated and reviewed for formatting (Pandoc + CSL usually handles it; if using Word, ensure all sources are listed and sorted). The **Index** should be alphabetically sorted and properly indented (sub-entries if any). Correct any glaring issues (like an index entry that points to many pages might need sub-entries for clarity, etc.).

The inaugural workflow concludes with a **fully structured, publishable manuscript** (complete with appendices) and all derivative materials ready for use. At this point, you have a *first edition* released, with an established system to produce updates. Next, we address the iterative workflow for maintaining and updating this content over time.

Iterative Workflow: Updating and Maintaining the Book Over Time

The iterative workflow outlines how to efficiently update the book and related outputs for new editions or ongoing revisions. The goal is to make targeted updates (new content or corrections) **without rebuilding everything from scratch**, using the modular structure and tools established in the inaugural process.

Stage 1: Plan and Scope the Update

- **Identify Update Triggers:** Determine why an update is needed. Common triggers:
- New developments or research in the field (e.g., a change in tax law affecting estate planning, or new case studies worth including).
- Feedback from readers that suggests clarifications or corrections (maybe a first edition reader noted a missing definition or a confusing section).
- The desire to add content (perhaps a new chapter in Part IV on emerging technologies in wealth management, or an expanded section on a particular discipline).
- **Scope the Changes:** List which chapters or sections will be affected. Categorize them by magnitude:
- *Minor edits:* grammar fixes, updating statistics, adding a reference – these don't affect structure.
- *Moderate edits:* rewriting paragraphs for clarity, adding an example – content changes but within existing sections.
- *Major edits:* adding a new chapter, removing or merging chapters, reordering sections, adding new figures, etc. These impact structure, numbering, and possibly the overall page count notably.
- **Branch the Project:** Using the version control set up earlier, create a new branch for this update (e.g., `edition_1.1` or `edition_2.0` depending on significance). This allows work on the update while preserving the published edition. In Git: `git checkout -b edition_2.0`. If not using Git, duplicate the project files into a new folder (but clearly mark versions to avoid confusion). This branch will encompass all changes for the new release.

- **Update Strategy:** Plan whether the update will be incremental (small changes rolling out perhaps as a revision notice) or a full new edition that will also need new ISBNs, etc. For a full second edition, you might also plan to write a new preface or author's note highlighting what's new. Add that to the outline if so.

Stage 2: Content Revision in Modules

- **Work Chapter-by-Chapter:** Thanks to the modular approach (each chapter in a separate file or section in Scrivener), you can go directly to the affected chapters and edit only those. For example:
- If Chapter 3 "Historical and Modern Lessons" needs an update with a recent example, open that chapter's markdown or Scrivener document and add the example (maintaining the same style).
- If a new Chapter 22 is to be added on say "Digital Assets in Wealth Transition," draft that chapter separately (perhaps start in a new markdown file `22_digital_assets.md` or add a new document in the Scrivener binder).
- If a chapter is removed or split, handle those files accordingly (e.g., delete or split into two files, and update the master list of chapters).
- **Leverage LLM for Drafting Updates:** For significant new content, you can again use AI assistance. For instance, if adding a chapter on an emerging topic, feed the AI a prompt with what key points to cover. Since the overall book context exists, you might even provide the AI with the updated outline and ask it to generate a draft for the new chapter (similar to Stage 2 of inaugural). Use "enrich outline" mode if using Youbooks with new source material, or simply use ChatGPT with specific instructions. Then edit the AI-generated addition to fit seamlessly with your existing prose.
- **Maintain Consistency:** As you update, ensure the terminology remains consistent. If a new term is introduced, add it to the Glossary file. If you changed a term (say you originally used "cadre" and now prefer "consortium"), do a project-wide find/replace and update accordingly in all chapters and the index. Consistency is easier to maintain with modular files—consider using a tool like **grep** or an IDE's search to find occurrences across all files.
- **Cross-Reference Updates:** If structure changed (new chapter or section), renumbering will happen automatically at compile time with our cross-reference system, *but* you should update any explicit mentions in text of chapters. For example, if you added a new Chapter 2, what was "Chapter 2" before is now Chapter 3, etc. Search the text for any hard-coded references like "in Chapter 7" and ensure they either use automated referencing or you adjust the numbering. Ideally, all such references were automated (`@sec:label`) so they will update themselves. If not, convert them now to automated ones to avoid this pain next time.
- **Figure Updates:** Edit the figures if needed:
- If a figure's data changed (e.g., you cited a statistic on wealth transfer that's now updated), update the figure image. This might be as simple as editing an Excel chart and re-exporting the PNG, or adjusting a diagram to add a new element. Save the new image with the **same filename** if it's essentially the same figure updated (this way, the compile will just pick up the new image). If it's an additional figure, create it and give it a new label and filename.
- If you insert a new figure in the middle of the book (say in a new chapter), all subsequent figure numbers will auto-increment on next compile (pandoc-crossref will handle it). The List of Figures will also update. Just ensure you provide a caption for the new figure.
- If a figure is removed, delete its reference from text and remove the file if not needed. The numbering again auto-adjusts. After removal, search for any stray mentions of "Figure X" that might have referred to it.
- **Bibliography and Citations:** Add any new sources to the `.bib` file and cite them in text as needed. Remove or mark obsolete citations if you removed content that referenced them. After changes, run

Pandoc with `--citeproc` to regenerate the bibliography – it will automatically include new references and drop those no longer cited. Always double-check the Bibliography after an update to ensure it's accurate.

- **Index Maintenance:** If you added significant content, consider new index entries. For example, new chapter topics or new case studies should be indexed. Add `\index{NewTerm}` in LaTeX or mark in Word for those new terms. If a section was removed, you might have orphaned index entries – scan the index after compilation or if using LaTeX's `.idx` file, check that no entries refer to nonexistent pages (the indexing tool usually just drops them if they never appear). You might need to regenerate the index from scratch by running the tool again (which you'll do in compile anyway).
- **Glossary:** Update the Glossary appendix with any new terms or modified definitions. Remove terms that are no longer used if a chapter was removed. (One way to catch this: if you maintain the glossary manually, run a search in the text for each term – if it's not found anymore, you might drop it or keep it for completeness if you think readers still benefit from the definition.)

Stage 3: Regenerate Affected Outputs

One advantage of our setup is that we don't need to handcraft each output again – we use our automation to regenerate the final products. However, not all outputs always need full regeneration depending on the changes.

- **Full Book:** Always recompile the full book PDF/DOCX. Even a minor text change requires repagination, so you'll produce a new PDF. Use the same Pandoc/LaTeX pipeline (perhaps updated with new metadata if edition number changed). Because of cross-references, it's safest to re-run the whole thing rather than trying to insert pages into an existing PDF. Given we have a script or Makefile from before, just run it again on the updated source. This will output a new PDF with updated ToC, lists, index, etc. Examine the log (if LaTeX) for any warnings (especially if index entries changed significantly).
- **Prospectuses and Summaries:** Determine if they need updates:
- If changes were **minor** (typos, small clarifications) that don't affect the overarching message, you *might* leave the one-page summary as-is. However, the full 5–10 page prospectus likely mirrors chapters; if any chapter had a notable change, update that chapter's summary paragraph. For example, if you added a new chapter entirely, you should add a summary of it to the full prospectus and possibly mention it in the one-page and exec summary if it's an important addition. If you removed a chapter, remove or adjust its summary.
- For a **new edition release**, it's prudent to update all summary formats to reflect the biggest changes. It's also an opportunity to refresh the wording. Use the LLM again if needed: e.g., "Summarize the new Chapter 22 in 2 sentences and integrate it into the existing summary text." Or "Update the one-page overview to include a mention of digital assets as introduced in the new edition."
- After editing the summaries, run Pandoc to regenerate their PDFs/DOCXs/HTML. These are quick since they're short. Keep version control of these as well (maybe include an edition notation on the summary if needed, like a footer "Updated 2025").
- **Workbook:** If the main content changes were significant, the workbook may need adjustments:
- New chapter added → consider adding a new exercise section to the workbook for that chapter.
- Removed topic → remove or rephrase the corresponding workbook activity.
- Any numerical updates or terminology changes should reflect in workbook text as well.
  After editing in Word/LaTeX, generate a new workbook PDF. If using Word, you might just edit the

original .docx and resave as PDF. Check all form fields still work after edits (sometimes editing text can accidentally delete a form field, so re-add if necessary in Acrobat or Word's form editor).

- **Parallel vs Full Reprocessing:** The modular approach means you can avoid regenerating content that didn't change (for example, figures that are untouched don't need re-export). But ultimately, the final compilation is often easiest done for the whole book at once. And since Pandoc/LaTeX will handle the entire manuscript, "reprocessing" all content is not a huge burden – it's automated. The key time saver is you didn't have to manually rewrite summaries or redo layout from scratch – you only updated pieces and ran the toolchain.

It can be helpful to use a table to decide which outputs to update given a type of change:

Change in Manuscript	Full Book PDF/DOCX	5–10p Prospectus	One-Page Summary	Exec Summary	Workbook PDF
Typos or grammar fixes	✔ Recompile	– (No change)	– (No change)	– (No change)	– (No change)
Updated data or minor content	✔ Recompile	✔ If affects key point in a summary, update that line	✔ Possibly tweak phrasing if the main message shifts slightly	✔ Maybe, if overall conclusion altered	– Usually no (unless workbook referenced that data)
New section within a chapter	✔ Recompile	✔ Update that chapter's summary paragraph	– Possibly not, unless it's a major addition worth mentioning at high level	– Unlikely	✔ Add to workbook if it's an actionable concept
New chapter added	✔ Recompile (ToC & refs update)	✔ Add new summary paragraph for it	✔ If it's a major theme, incorporate mention	✔ Yes, if it influences main conclusions	✔ Add new exercise section
Chapter removed/ reordered	✔ Recompile	✔ Remove or adjust summary text	✔ Possibly not (one-page may not have mentioned it explicitly)	✔ Maybe slight rewording if needed	✔ Remove related section
Terminology change (e.g., rename a concept)	✔ (global text change)	✔ (reflect new term)	✔ (if term appears)	✔ (if term appears)	✔ (update term in exercises)

Change in Manuscript	Full Book PDF/DOCX	5–10p Prospectus	One-Page Summary	Exec Summary	Workbook PDF
New figure or figure update	✔ Recompile (figure list & refs)	– (Summaries typically don't include figures)	–	–	– (Unless workbook had a diagram referencing it)
Bibliography update (new sources)	✔ Recompile (new refs in bib)	–	–	–	–

(✔ = regenerate/update; – = no need to update content)

Stage 4: Finalizing the New Edition

- **Integrate Changes and Review:** Merge the changes on your branch into the main line if you want (or keep the branch as the published version). In Git, once all is done and tested, you might tag this as `v2.0` and optionally merge back to `main`. This ensures the main branch is up-to-date for any further small errata fixes. Keep the old tag for reference.
- **Proof and Quality Check:** Just like with the first edition, do a complete run-through of the updated book PDF. Check that all updated parts read well in context. Pay attention to any newly added content for layout issues (e.g., a new chapter starting on an odd/even page as desired if using book formatting, page breaks looking okay). Verify the table of contents reflects new chapters and that index entries for new content are present. Ensure no "??" reference placeholders (which can happen if a label was mistyped). If you see any, fix the source and recompile.
- **Update Web and Publishing Platforms:**
- If the book is available via a website or WordPress, update those pages/posts. For the summaries on WordPress, you can simply re-import a new WXR or manually edit those pages. Since WordPress might treat an import as new pages, you may choose to manually copy the updated summary text into the existing pages (to retain the same URLs). Alternatively, import and then delete the old versions. Ensure to update the date or add a note "(Updated [Month Year])" on those pages for transparency.
- If you have a blog or an announcement area, you might post about the new edition's changes for readers.
- For distribution: upload the new PDF to wherever the old one was (KDP, etc., if doing a new edition on Amazon, follow their process for new edition or simply replace the content if minor revision). Provide the new workbook if that changed.
- **Maintain Backwards Compatibility:** If some readers still have the old edition, consider providing a brief changelog or at least ensuring the new edition's introduction or preface notes what has been updated or added. This is more content work, but useful in professional works. For example, add a paragraph: "**What's New in the Second Edition:** This edition includes a new chapter on digital assets, updated statistics post-2025, and an expanded glossary."
- **Continuous Improvement Setup:** Now that the process has run end-to-end, you might further streamline:

- Set up a continuous integration (**CI**) pipeline (e.g., GitHub Actions) that automatically compiles the book and summaries to PDF whenever you push changes. This way, you always have an up-to-date draft PDF to review and it catches errors early.
- Consider writing small scripts for repetitive tasks. For instance, a script to regenerate all summaries from the book content using an AI API could be developed, or a script to scan the manuscript for undefined terms to add to the glossary. These aren't required, but can speed up future iterations.

Stage 5: Ongoing Collaboration and LLM Support

- **Collaboration:** If more contributors join (co-authors, editors), use Git branching and merging to manage contributions. Each person can work on a feature branch for a chapter and merge changes in. Resolve conflicts using Git's tools. This ensures the iterative process can scale with a team.
- **LLM for Editing Cycles:** Continue to utilize large language models for iterative editing tasks:
- Use them to periodically *summarize each chapter* anew to see if the AI picks up any inconsistencies or filler text – if the AI summary misses a key point, perhaps the chapter didn't emphasize it enough, which is a cue for further editing.
- Use them for *sensitivity or clarity reading*: e.g., "Is there any sentence in this chapter that is confusing or could be misinterpreted?" The model might highlight areas to refine.

- As the field evolves, you might even use AI to monitor news/research. For example, an AI with access to current info (or via your provided sources) could alert you when there are updates in estate law or new significant publications in family wealth management, which you can then incorporate in the next revision.

- **Modular Updates Without Full Re-run:** In cases of extremely minor interim updates (like correcting a typo or adding a single reference), you might not go through the whole formal release process. For instance, you can directly edit the WordPress pages for the summaries with the small tweak, or issue a corrected printing of the PDF via the script (changing version number slightly to 2.1). Because of our setup, even a one-line change still calls for recompiling the PDF to keep everything consistent (page numbers etc.), but since it's automated, that's trivial. **Key point:** The heavy lifting (outlining, initial drafting, toolchain setup) from the inaugural workflow pays off by making iterative changes relatively quick, isolated, and low-risk.

Stage 6: Preserve and Branch for Future

- **Archive Old Editions:** Keep tags in Git or dated copies of outputs for each edition. This is important for record-keeping, especially for a book that might have citations or page references used elsewhere; you want to be able to trace if someone references "page 100 of first edition" vs second edition. Since Git already tracks, you can always regenerate the first edition from the tag if needed.
- **Branch for Experimental Changes:** If you foresee larger changes (e.g., a possible spin-off book or a drastically restructured third edition), create a branch for those experiments. For example, maybe you want to tailor the book for a different region or a different but related topic (like a version focusing solely on philanthropic transitions). You can branch off and modify content without affecting the main line. This branching approach is akin to maintaining the "source of truth" while exploring new derivatives, much like software version branches [30] .
- **Repeat the Iterative Cycle:** Each time changes accumulate, repeat Stages 1–5. The process becomes faster as you become familiar with the toolchain and as the amount of new content per edition usually decreases relative to the initial creation. Always aim to integrate new content in a

modular way to ease future updates (e.g., if adding a case study, perhaps put it in a call-out or its own subsection, so that in future it can be updated independently).

By following these comprehensive workflows, the project *Founding an Ecosystem for Wealth Transition* can be created and maintained efficiently. The inaugural workflow ensures a **well-structured first edition** is produced with full appendices and multiple audience-specific outputs. The iterative workflow leverages the initial setup (modular content, automation, and AI assistance) to allow smooth, partial updates and new editions without redoing the entire effort. This combination of **author-driven planning** and **LLM-supported execution** – alongside tools like Youbooks, Scrivener, Pandoc, LaTeX, and scripting – results in a sustainable system for book publishing. The content remains consistent across formats, figures are handled despite the text-only core platform, and future changes can be managed in a controlled, efficient manner.

Overall, these workflows turn what could be a daunting one-off book project into a maintainable, evolving knowledge product – an ecosystem of its own, for content creation and knowledge dissemination.

Sources:

- Youbooks AI non-fiction book generator description [11] [10]
- Pandoc for multi-format conversion and cross-referencing [7] [18]
- Git for version control in writing (single-folder project, easy branching) [17]
- WordPress WXR format for content import/export [27]

---

[1] [2] [3] [4] [6] [13] [16] youbooks_prompt.md
file://file-R2539Xtun6deENZFVv58Bs

[5] [11] [12] [15] FAQs
https://help.youbooks.com/en/category/faqs/article/faqs

[7] [20] [22] [25] [26] scrivomatic | A writing workflow using Scrivener's style system + Pandoc for output...
https://iandol.github.io/scrivomatic/

[8] Youbooks: The Best AI Book Generator That Works : r/WritingWithAI
https://www.reddit.com/r/WritingWithAI/comments/1h57mno/youbooks_the_best_ai_book_generator_that_works/

[9] [10] [14] Youbooks: The Best AI Book Generator That Works - 2025
https://webmoneyai.com/product-review/youbooks-best-ai-book-generator/

[17] tools - What's the best version control system for a book writer? - Writing Stack Exchange
https://writing.stackexchange.com/questions/7249/whats-the-best-version-control-system-for-a-book-writer

[18] [19] [23] [24] latex - How do I make a reference to a figure in markdown using pandoc? - Stack Overflow
https://stackoverflow.com/questions/9434536/how-do-i-make-a-reference-to-a-figure-in-markdown-using-pandoc

[21] [30] Git for writers: Write fiction like a (good) programmer | by Vanessa Guedes | Medium
https://medium.com/@vanessainpixels/git-for-writers-write-fiction-like-a-good-programmer-ea6f0309a69a

27 The WordPress eXtended Rss (WXR) Export/Import, XML Document Format Decoded and Explained. – The Developer's Tidbits
https://ipggi.wordpress.com/2011/03/16/the-wordpress-extended-rss-wxr-exportimport-xml-document-format-decoded-and-explained/

28 forms - Creating fillable PDFs - TeX - LaTeX Stack Exchange
https://tex.stackexchange.com/questions/14842/creating-fillable-pdfs

29 How to Create Fillable PDF Form for Free using Latex - YouTube
https://www.youtube.com/watch?v=KFLu5DPt16E