Links

These docs: https://drive.google.com/drive/u/0/folders/0B6xsrc5BVkagVjl3WGFxSEJoV0E
Schedule: https://groups.drupal.org/node/516460

(Note: This didn't happen at Dev Days, but meh. :))

Discussion lead

- Tim Plunkett

Attendees

- Adam Hoenich
- Alex Bronstein
- Angie Byron
- Emilie Nouveau
- Damien McKenna
- Jakob Perry
- Kris Vanderwater

Goals

- Come up with consensus for next steps for the layouts in core initiative
- Find another initiative coordinator: Emilie [Fixed https://www.drupal.org/node/2867599]

Issues for discussion

-

Prior art

-

Notes

- Where we're currently at:
    - Lowest common denominator: Panelizer and Display Suite and put it in core as Field Layout experimental module (regions for entity displays per-view mode, per bundle)
    - Pretty much done.
- What the ideas are:
    - Replacing theme-declared regions in info.yml with a layout. Right now, theme has a single layout, as a list of regions. **Proposal**: instead of a list of regions in info.yml, ship with one single layout.yml file as MVP. (Themes can opt in to

providing a single layout that's in use on every page that it's rendered on.) ← this code is basically done, "gateway" to more complex use cases. https://www.drupal.org/node/2811179
        - Doesn't empower site builders, but is a step toward unifying APIs, creating one way to do region definition. But, might have render pipeline implications.
        - This is a nice DX improvement, but not really part of the Layouts initiative.
    - In Field Layout module, use proper blocks (possibly renaming to “component”, from current EntityDisplay internals) instead of whatever's currently being used (DX).
    - Or, go "full" Panelizer in core. One-off layouts for a specific entity and defaults. (Landing page functionality)
    - Panels vs. Page Manager vs. Panelizer have walls between them, but they are largely only necessary for contrib. In core, there should be unification.
    - Emilie's mockups: Add > Landing Page > [Settings Tray] Enter title / path > Choose Layout. > [one row] > Populate with content. [Optional: Add another row]
        - Panelizer of content area, not of entire page area (Panels Everywhere)
        - How to deal with themes that define sidebars? Checkbox on landing page form: "Disable sidebars"
    - Page.html.twig "just works" as a layout (!). Removes DX concern of there being two systems for defining regions.
    - But also, want to be able to swap layouts, for example on the "About" page, use a right sidebar. How do we replicate page--xxx--yyy.html in the UI?
    - Provide a special {{ xxx }} when a layout is desired; this way themers have control.
    - Goal: **Empower site builder in a way that doesn't hamper the themer** (allow them to override/enhance).
    - Missing functionality from Field Layout: Selection Rules. The ability to have more than one layout for a single view mode, nor for all pages except this one have the same layout.
    - Timeline: 8.5 (about 9 months from now)
    - Key requirements:
        - A *good* IPE (possibly via Settings Tray?)
        - A way to make landing pages (must be content entities, possibly a node type)
        - Making sure your landing page actually saves. ;)
        - Panelizer supports revisions. (Workflow stuff)
        - Right now, Panelizer supports revisions, and Page Manager supports exporting and deploying via version control.
        - Page Manager use cases: overrides (e.g. Search page override), building "app" e.g. a “how-to” wizard. Use Page Manager like Views: use it as wrapper around all admin pages, UI around routing. No editorial workflow,

        - don't give access to Page Manager. Anything with editorial workflow == Panelizer.
    - Panels Everywhere not seen as a "core" use case. (13K installs vs. 36K installs of Panelizer)
- Our hit-list of what's next (consensus)
    - Entity-based focus is still the most important (Panelizer in core). **The ability to override node/5.** (Only on content types for which layouts are enabled.)
    - User interface for content authors to be able to create a landing page, choose a layout, populate the layout. (Usable in either Page Manager or Panelizer.)
        - Use same UI on backend for pre-populating Default layout; need Lorem Ipsum default content, because don't have node/5's content.
    - Populate layout with fields, content, blocks, etc. [Broadest definition of "content"]
    - 1 default layout per bundle per view mode. [Multiple defaults is a contrib thing.]
    - Should be able to add entire other entity, Powered by Drupal block, body field. [These can all be done in parallel; smooth over differences in UI.]
        - Fields are placeable, they have a required context of an entity. [In context of Entity Displays -- "Components" internally, convert to Blocks]
    - Need to ensure that this doesn't introduce backwards-incompatibility into contrib. [Need Migration path, or known "fences" around areas.]
    - Move CTools Blocks / Entity Block into the Blocks system. [Not directly related to Layouts, per se.]
    - Basically: Need to be able to replicate Panels, CTools Blocks, some type of backend admin UI for contexts [8.6+], and (parts of) Panelizer (sans add arbitrary context UI, multiple defaults). Display Suite is Field Layout++. Panelizer becomes the same over time, the more of it moves into core.
    - Core needs to support multiple defaults behind the scenes in the API. (No UI for it.)

- What the challenges are:
    - Variants at the page level.
    - Not just about swapping layouts, also about how routes map to layouts.
    - Instead of {{right}} {{left}}, in Twig templates, just {{content}}. Themes become just a skin, for page layouts provided elsewhere.
    - The point of this initiative is not to remove the Block module. Block UI would switch between layouts instead of themes, when showing regions. Or, could move to a wizard: Add > What layout? > What region?
    - If we make radical adjustments to Bartik, what do we do about people who are learning Drupal? The last 18 months of training materials on how to write a Drupal 8 theme. New themes? Move legacy themes to contrib?
    - What about Forms? (Form arrays present huge challenges.) Page Manager and Display Suite allowed changing entity forms. DS doesn't do this anymore. Field Layout by accident does. :) Node form in D8 already does 2 columns… Why not use Field Layout for this?

        - These layouts require presenting everything they're given vs. being able to choose. UI challenges.
        - Ideally: *everything* has a layout; never have to worry about an entity form never having a layout.
    - What to call "page elements" / "page components" / "blocks" / "modules" / "widgets"

Action items

- **140-character summary in "plain English" :)**
- Create a meta issue plan laying out the next steps.

Tips for discussion leads

1. Set an agenda in advance (and help people stick to it).
2. Ask someone to be the primary notetaker (other than yourself) and ask everyone to help collaboratively take notes.
3. Make sure that all results of the discussion are documented publicly and shared in the Drupal issue queues, etc.
4. Pick a quiet place for an initial, focused, timeboxed meeting, then continue your discussion in the public sprint room after the meeting.
5. Limit your discussion participants to about 10 at most for a productive meeting.
6. Don't discuss for more than 60-90 minutes in a row. People need breaks to be productive.
7. At the end of the meeting, summarize how the discussion goals were covered (or not).
8. Identify action items and who is responsible for ensuring they happen.