Design and Build User-friendly vs. SharePoint User-friendly Solutions

Wes Preston
PWR202

Abstract

- Audience: Info Workers, Dev

- Solution and application sites built in SharePoint or O365 often rely on a baseline understanding of how SharePoint works. This entry point can stifle user adoption and satisfaction. We can, however, design and build solutions that align more tightly with users' needs using SharePoint's out of the box capabilities and simple client-side rendering techniques.

- In this session we'll walk through a common example and show the advantages of web-user-centric design.
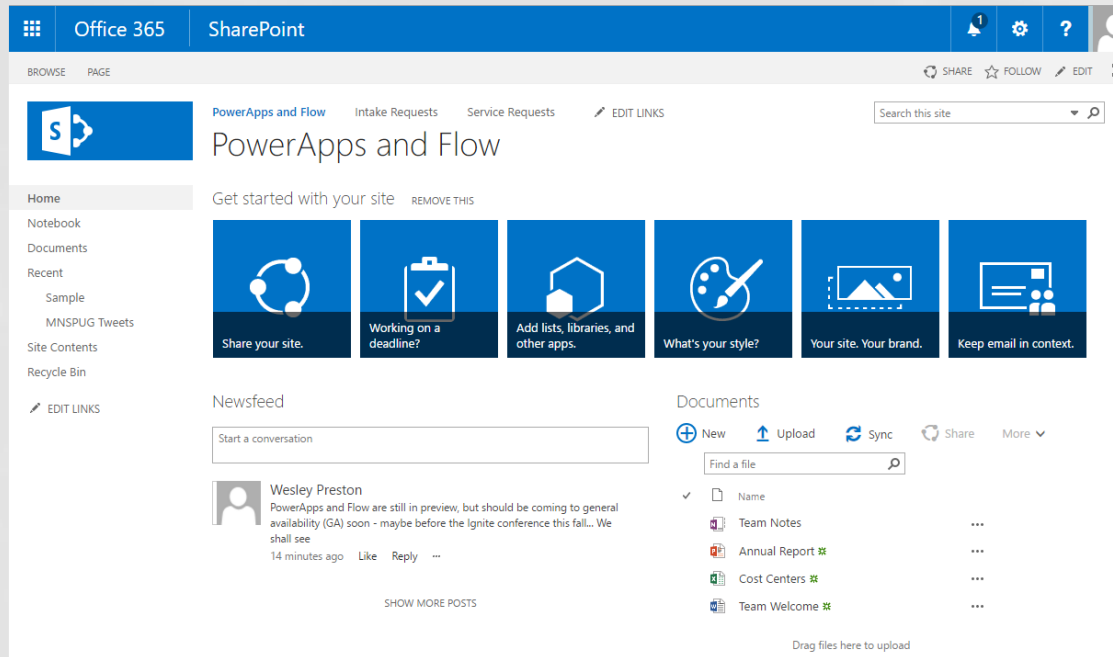
Wes Preston

- Owner / Principal Consultant - TrecStone
  - Based in Minneapolis, MN
  - MVP – SharePoint Server (2009 -)
  - SharePointing since 2003
- Contact:
  - www.idubbs.com/blog
  - @idubbs

Why are we talking about this

- Organizations want to get more out of current investments (SharePoint, O365, etc.)
- Microsoft's push to power users
- Business solutions that don't require development, or less of it
- Users report not enough training (reference)

What are we talking about...

- Sites for collaboration: Do a great job

- The 'new experience': even better.

From this:

To this:

Challenges:

- Accessibility of the list
  - Where is it? Part of a team site?
  - Are permissions set up correctly?
- Lack of instructions on how to use the list
  - The trend is to simplify… break it down to easy options, steps, targeted data, etc.
- Can't find their own requests easily
  - What happened to my request?
- Folks managing the list aren't working effectively
  - Oh, didn't see that one request…

Benefits:

- Role-based experience
- Simplified and/or enhanced views
  - "Simplified": Still use views, but put more thought into them
  - "Enhanced": CSR to make them more powerful
- Well documented
  - Use SP's capabilities to build doco right into the site
- Easily supported
  - Self-document the solution using SP's features

Benefits:

Why is this approach so useful?

- It works really well for non-office workers
- It works really well when there are a significant number of users without SharePoint training or experience

- It works well for SharePoint people too…

Tips on Design

- Whatever works in your organization…
    - Agile vs. Waterfall, etc.
    - Best to be iterative – users WILL want changes

- Agile: User Stories
    - Helps identify roles / user types

- Wireframing: Balsamiq

- Maybe this is just a style guide, tips and tricks for best practices

- When we're done, you tell me…

Let's get to it.

Understand the use cases

- Identify the people and roles involved
- Work through 'user stories' for each role

- As with 'Agile' approaches, the site will likely evolve over time

- Once site is created, it's a good time to start documentation... even just placeholders

Site Taxonomy

Where should the solution be built?

- It's own site collection?
- A sub web of another site?
- Within an existing site?
- Cloud or on-premises?

Important considerations:

- Users and access

Example: Service Request

- Lots of people outside the team need access
  - Part of a service team that likely has a public-facing site already
- Not overly complex, so *could* be built into a site
- Prefer isolation – so either site collection or sub web
- Might want to put on O365 for easier mobile access… (if hybrid)
- Demo: Sub site

Identify User Stories and Roles

- Who needs this solution?
- Who's creating / adding items
- Who needs to view items?
- Who's updating items
- What kinds of notifications are needed?
- Who needs access to the whole process/site?

Example: Service Request

- Requester
  - Submit requests
  - See status of requests
- Persons doing triage on incoming requests
  - View request statuses
  - Assign request to someone
- Persons responding to requests
  - See what requests have been assigned to me
  - Work on assignments
- Management
  - See statuses of requests
  - See any reporting
- Dev/SharePoint team

Lists and Views

- Create core lists
- Create any lookup or other lists

- Start building out core views
  - Pay attention to which views you want to show up in Views dropdown, vs. one-off views per web parts

Example: Service Requests

- Create the 'Requests' list
- Create any lookup lists (departments, request types, etc.)

- Later we'll use SPD to hide the lists
SP knowledgeable folks could still figure out URLs – this isn't security, it's just locking down the UX
  - Don't want users distracted by the default list views

Site Pages

- Create a folder for each role*
  - May need to change library settings for this
- Later: Set security at the folder level

- Home page for each role
- Page per key functionality
  - Allows for clear actions and on-page instructions

Site Pages – Tips

- Create placeholder pages
  - Will help see how solution is being built out

- Change Site Pages settings to show in Quick Links
  - Just easier to get to pages more quickly while building out
  - Remove from Quick Links before launch

Example: Folders

- ## ManageReqs
  - Could be separated by triage and workers
- ## SiteOwners
  - Could be separated between business owners and SharePoint Team
- ## Management
  - Generally only give as few permissions as possible

Site Pages

⊕ New   ↑ Upload   ⟳ Sync

All Pages   By Author   By Editor   ...

✓   ☐   Name

📁   SiteOwners   ...

📁   Management   ...

📁   ManageReqs   ...

🖻   Home   ...

🖻   How To Use This Library   ...
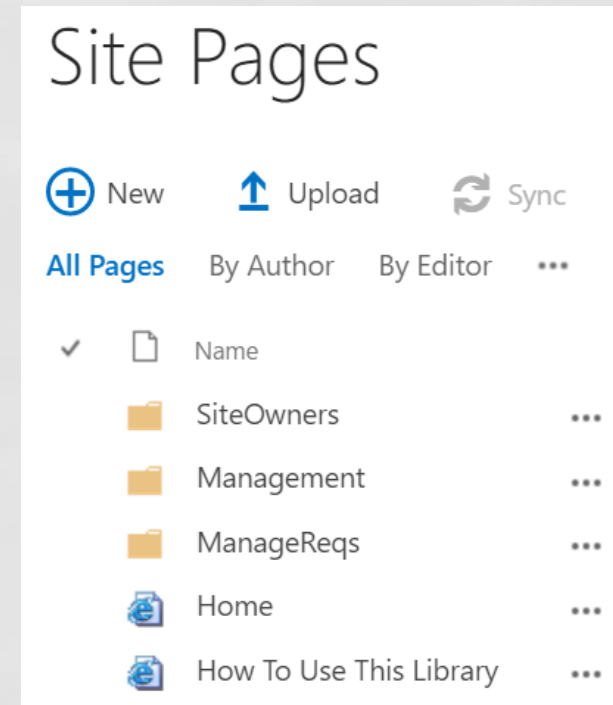
Example: Pages

- Requesters – 'Normal' users
  - Home page is site home
    - Create new requests
    - See status of open requests
  - My Requests Page
    - See open requests
    - See historical requests
  - Instructions / Information page
    - How to create, view, etc.
    - Who to contact with questions, set expectations, etc.

Creating Pages

- Add text description of what the page is for.
- Add the app part
- 'Help' links

App Part Configuration

- Select an existing view or create a view
- Change the app part title: 'My Service Requests'
- Change the link from the app part Title
  - We don't want users navigating to normal list views
  - Return to current page

□ Advanced
☑ Allow Minimize
☑ Allow Close
☑ Allow Hide
☑ Allow Zone Change
☑ Allow Connections
☑ Allow Editing in Personal View

Title URL

/sites/PAFDemo/ServReqs/Lists   ...

Example: Pages

- ManageReqs
  - Home – 'Triage' Page
    - New, unassigned requests – to be assigned or 'grabbed'
    - Open but incomplete requests
  - My Work Page
    - Filtered by current user to see what is open and assigned
  - Link to 'normal' list views
    - Default to All Items

  - Links back to home

Example: ManageReqs Home

- Aka: Team Home

Example: Pages

- Management
  - Home
  - Separate views / report pages as needed
    - Group by summary view
    - If require something like monthly numbers will need to add new fields for year and month to group on

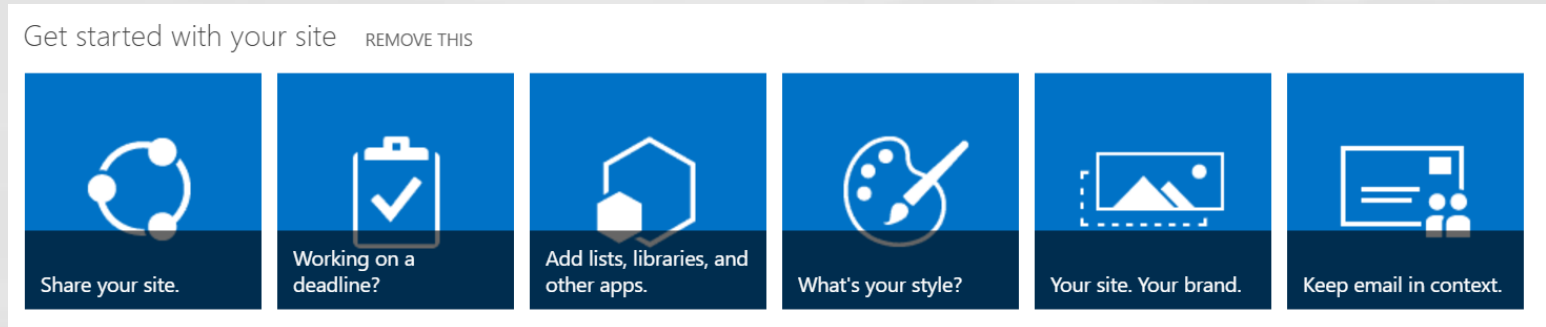 - Folks working requests may want access to these views / reports

Example: Pages

- SiteOwners
    - Home - Links to all hidden lists for maintenance
    - About (opt) – Overview of solution, history, etc.
    - About Dev (opt) – as needed for development notes
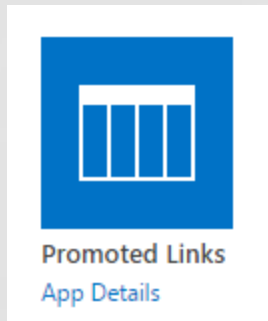
Promoted Links

- Microsoft has found them to be effective, so let's use them too:

Get started with your site   REMOVE THIS

Share your site. | Working on a deadline? | Add lists, libraries, and other apps. | What's your style? | Your site. Your brand. | Keep email in context.

- Audience targeting
  - Apply security to list items - mainly Home page
- Create separate lists for each *role* homepage

Example:

- Remove the default 'getting started' part
- Create a new one, add a few placeholders, and drop it on the home page

Security

Set up:
- SharePoint group per role

- Apply group access to lists by usage
- Apply group access to folders in site pages
- Apply group access to items in main promoted links

Example: List Security

- Service Request List
    - Generally pretty open
    - May limit users to create – not edit

- Departments list
    - View only for users

Example: Site Pages Security

- Share With - Advanced

Example: Promoted Links Security

- Key to audience-targeting home page:
  Set permissions on individual items in the
  Promoted Links (NavHome) list

- Note: Don't need to lock down permissions to
  separate promoted links lists – only seen by
  group that can see the page anyway

Example: Promoted Links Security

- As seen by team users:

Example: Promoted Links Security

- As seen by request users:

Forms, Content Types, Custom

- What fields are needed on each form?
- Are additional forms required?

- Content types… for new items or 'Sarah' method…

- Custom JS or other forms… Stratus, etc.
- CSR nice for linking access to additional forms

Example:

- New request shows only limited fields
- User view – limited fields (custom form?)
- View for person working issue – all fields (use the default form?)

Web parts, Views, CSR/JSLink

- Not *required* for MVP…
- Anything to add to views to enhance the user experience?

Example:

- Conditional formatting opportunities
    - Triage page – unassigned
    - Reporting page – not closed after X days
- Replace view and edit links for main user view
- Change or remove Lookup field links
- Change 'no items' message for web parts

Notes: Workflows

- Certainly have an opportunity for automation but should also be reviewed with Microsoft Flow in mind

Example: Workflow

- None needed in this scenario

- Optional: Content type switch on new
  See references for 'Sarah's approach'

- Optional: Notifications
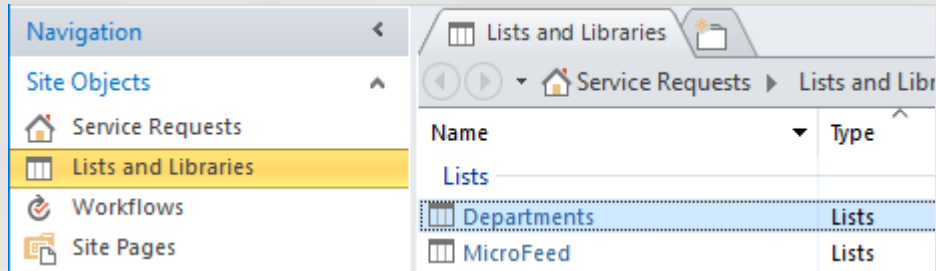 - New requests
  - Overdue requests

Cleanup

- Top and left navigation menus
  - Remove Site Pages (was added at start)
  - Clean up 'Recent'
- Use SPD to hide lists
- Test
  - Check web part headers on site pages
  - Navigation after forms
  - Revisit views with test data – filtering

SharePoint Designer – Hide Lists

- Not needed for non-SharePoint folks

- Useful for SharePointy people that like to poke around the site

- Also hides from folks managing the site
  - Add links for them

SharePoint Designer – Hide Lists

Page Cleanup

- Remove unneeded space and app parts
- Possibly use different page layout

Review: Documentation

In general, consider each role:
- User documentation
- Process owner documentation
- Management documentation
- Dev documentation

Use site pages or documents

Finished Product

Next Steps - Extras

- Solution Launch

- PowerApps
- Flow

PowerApps

- User App: New Requests and My Requests
- Manage Requests App: Triage and status

Note: Blog posts coming on these soon…

Microsoft Flow

- Notifications
    - If a request is of a certain type and priority, send specific notifications out
    - If a request is open for too long escalate by sending out different notifications

This specific scenario didn't have a lot of opportunity for Flow.

Summary and Notes

- Make actions clear and accessible
- Display targeted information
- Remove control confusion

- New page authoring and analytics coming to O365 will play a part
  - Easier to build pages
  - Better analytics for seeing if and how site is used

References

- Blog
  http://www.idubbs.com/blog/

- It's not SharePoint, it's the People. :P
  http://www.cmswire.com/digital-workplace/the-problem-with-microsoft-sharepoint-people/

- Modify Forms Using Content Types – Sarah Haase
  http://blog.splibrarian.com/2011/03/21/using-content-types-to-modify-the-newform-aspx-and-editform-aspx-pages/

Thank you for attending!

Wes Preston
@idubbs