1. Introduction to requirements?
   1. What needs to be designed/built?
      1. Cannot be gathered
      2. What users need to be able to do with site
      3. Functionality/content/data
      4. Questions not solutions
2. Generating Requirements
   1. A website’s feature set is determined by requirements. The more features we decide to implement, the more details we need for:
      1. What those features are
      2. How they should be designed
      3. What level of coding, scripting, plugin or CMS platform is necessary to implement them
   2. The balancing act
      1. Its all about us vs what’s in it for me?
      2. For existing sites, current users.
      3. For a new site, a specific target audience who may use it
   3. Requirements also come from stakeholders:
      1. What they believe will satisfy customers
      2. What they believe will deliver ROI back to their organization
3. There are 3 kinds of requirements things people:
   1. Say they need
      1. We make very confident but equally false predictions about our future behavior
         1. There is a huge difference between imagining how you’d use something and actually using it
         2. Our preferences at any given time are influenced by our emotions, our health and our environment
   2. Actually need
      1. When we have a problem, it can be effortless to imagine a solution that will solve it. However:
         1. That solution will often address a symptom instead of the underlying problem
         2. That symptom is the right place to start but it needs to be qualified and validated
         3. Play it forward to uncover the other actions, processes and motivations attached to the problem
         4. Do we fully understand every step in every task flow, from the standpoint of both the user and every other user role?
         5. Do we have any evidence that tells us this feature or function would solve a frequent or recurring problem?
         6. Is there research can we perform?
         7. Are there specific scenarios or tasks where user observation would help to fill in the blanks?
   3. Don’t know they need
      1. “People don’t want quarter inch drills. They want quarter inch holes.”
         1. The desired outcome is more important that the chosen tool
      2. Conversations about expected outcomes instead of features, function and content, produce more relevant requirements for … features, functions and content!
      3. When problems arise, we look for convenience first, which will almost always trump best practice.
         1. so instead of talking about the website or its features or functions, focus on:
            1. the problem people are facing
            2. what the desired outcome is
            3. what they’re most likely to do to solve it with or without your website!
4. Documenting Requirements
   1. **NEED TO DOCUMENT THEM**
   2. There is a common misunderstanding that following a lean or agile methodology means no documentation of any kind. **This is simply not true**
   3. It means **less** documentation not no documentation
   4. Documented reference gives the entire team, yours and the client’s two things:
      1. A sense of the scope of what you’re building
      2. A reference point for progress during the website project lifecycle
   5. Shared documentation also gives you a common way to label and escribe things in a manner everybody understands
   6. Eliminates excuses
      1. If decisions are documented and shared, anyone, at any time can pull up that document and get educated.
      2. You don’t have to write a novel; just capture it somewhere, somehow.
      3. Make sure everyone involved has access to it.
      4. As requirements are approved, that approval is noted as well