Set Yourself Up For Success

So you're finally on a project, your project manager assigns you some tickets. I know I know, you want to just get your nose into the code. I get it code is awesome! But you can save yourself a lot of headaches and meetings if you spend a little time up front setting yourself up for success.

Know Your Ticket

I know it seems obvious, but it needs to be said #READTHETICKET take some time read the title, read each part of the ticket. Read through it as many times as it takes for the details of the ticket to sink in. If any part of the ticket makes you say "Well that does not make sense…" or "I wonder what they want to happen in this situation" you should stop. Determine what details are missing from the ticket and add them to the ticket comments. Then reach out to your project manager, project lead to update them.

In the case of a feature the key questions to ask yourself.

- Does the ticket contain all specifics required for development
    - Wireframes.
    - Demo Videos.
    - General Screenshots.
    - Data Files (XML, JSON, METADATA etc.).
    - Feature specifications.
        - Styling info (font, color scheme, sizing , etc.).
        - Functional flow (customer clicks button and modal opens).
        - Mocked up data (XML, CSV, JSON , etc.) .
- What does the feature do?.
- What data is involved? (A2C functionality utilizes the product object).
- What is the end result?

You don't even have your IDE open yet and you have already defined the core concept of what this functionality will be. You have also done a service to your client by helping them create good solid requirements which will in turn create awesome code, great documentation and amazing features. The code that will create this feature is important but a feature can be created in concept without any code written. With our core functionality locked down and goals set you can now focus all your efforts on the fun part….THE CODE!

In the case of a bug fix the key questions to ask yourself.

- Does the ticket contain all specifics required for development
  - Videos with steps to reproduce the bug.
  - Screenshots.
  - Corrupt Data Files (XML, JSON, METADATA etc.).
- What should the functionality be doing in working condition end to end.
  - Does it effect any visual components of the site.
  - When reproducing are there console errors (browser or BM).
  - Does this bug effect any external facing feeds or data transfers.
  - Is the data coming in from a third party.
    - Is this data coming from a service call.
      - Is the service configured in BM.
      - What scripts are utilized to create these requests.
      - What is the entry point where this request is initialized.
    - Is this data coming from a file feed.
      - Are we sending data out or is data coming in.
      - Where does the file live (SFTP, AWS, webDav).
- What is the functionality currently doing end to end.

Again without even touching a keyboard we have set ourselves up for success! In front of you now you have all the puzzle pieces to play the game we all love…DEBUG THAT CODE!

Don't Guess When It Comes To Tickets

Clients want awesome features and they want them as fast as you can code them. The instinct can be to push forward with spotty requirements or some open questions in an effort to keep forward progress. The trouble with this approach is you often end up with discrepancies between the clients expectations and what you are able to produce; to avoid this you should always note and voice any questions or concerns. Part of your job as a developer is to assist your clients and support their vision; the other part of the job is to use your knowledge and the wealth of knowledge available at Red Van Workshop to create complete effective tickets.