Visual Studio 2017 Angular Template vs Angular CLI

1. VS 2017 deployment to azure as simply select deploy to azure. Site deployed an optimized. Other options include file and iis. They appear to work.
2. VS 2017 allows devs to make many changes without restarting the app. Previously it was necessary to restart for virtually all changes. Now VS behaves more like node.js dev.
3. VS 2017 error when debugging in IE. Runs fine in IE if copy link but not debugging.
4. Debugging not working on my machine. Worked on other devs.
5. VS 2017 when adding new items do not use AngularJS file types that appear in the list. These are for Angular 1 and Angular 4+ will need typescript files. Angular cli commands may or may not work.
6. Different File Structure used in VS vs Angular CLI
   1. VS ClientApp folder = CLI src folder
7. Different boot strap / start up
   1. Angular CLI src/main.ts used to bootstrap app
   2. VS clientapp/boot.browser.ts and clientapp/boot.server.ts
8. App Module
   1. CLI – src/app folder
      1. app.module.ts – standard angular approach
   2. VS – ClientApp/app folder
      1. app.shared.module.ts – same as app module in cli
      2. app.browser.module.ts – imports app.shared.module.ts not sure usage assume ms optimization
      3. app.browser.module.ts – imports app.shared.module.ts not sure usage assume ms optimization
9. Component files
   1. CLI – normal convention to store components under src/app/
   2. VS – ClientApp/app/component directory – additional file level added
10. App Component
    1. CLI – located src/app folder along with app module.
    2. VS – located ClientApp/app/components/app – component separated from module
11. Changing the selector for the angular spa
    1. Angular CLI uses my-app, which can be replaced by changing in the landing page and the app component.
    2. VS uses “app” and requires changing in index.cshtml (landing page), app.component.ts, boot.browser.ts and boot.server.ts.