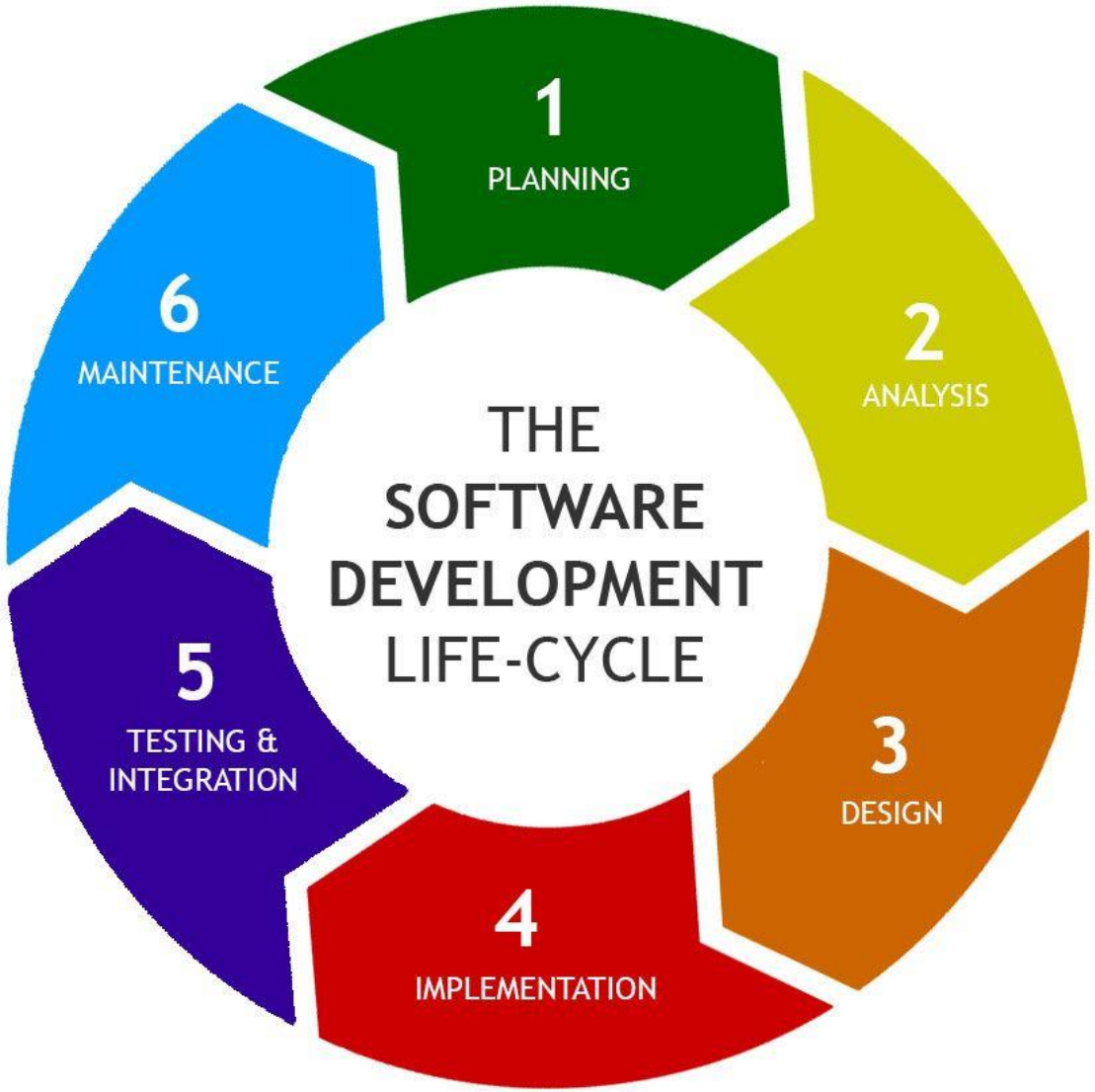
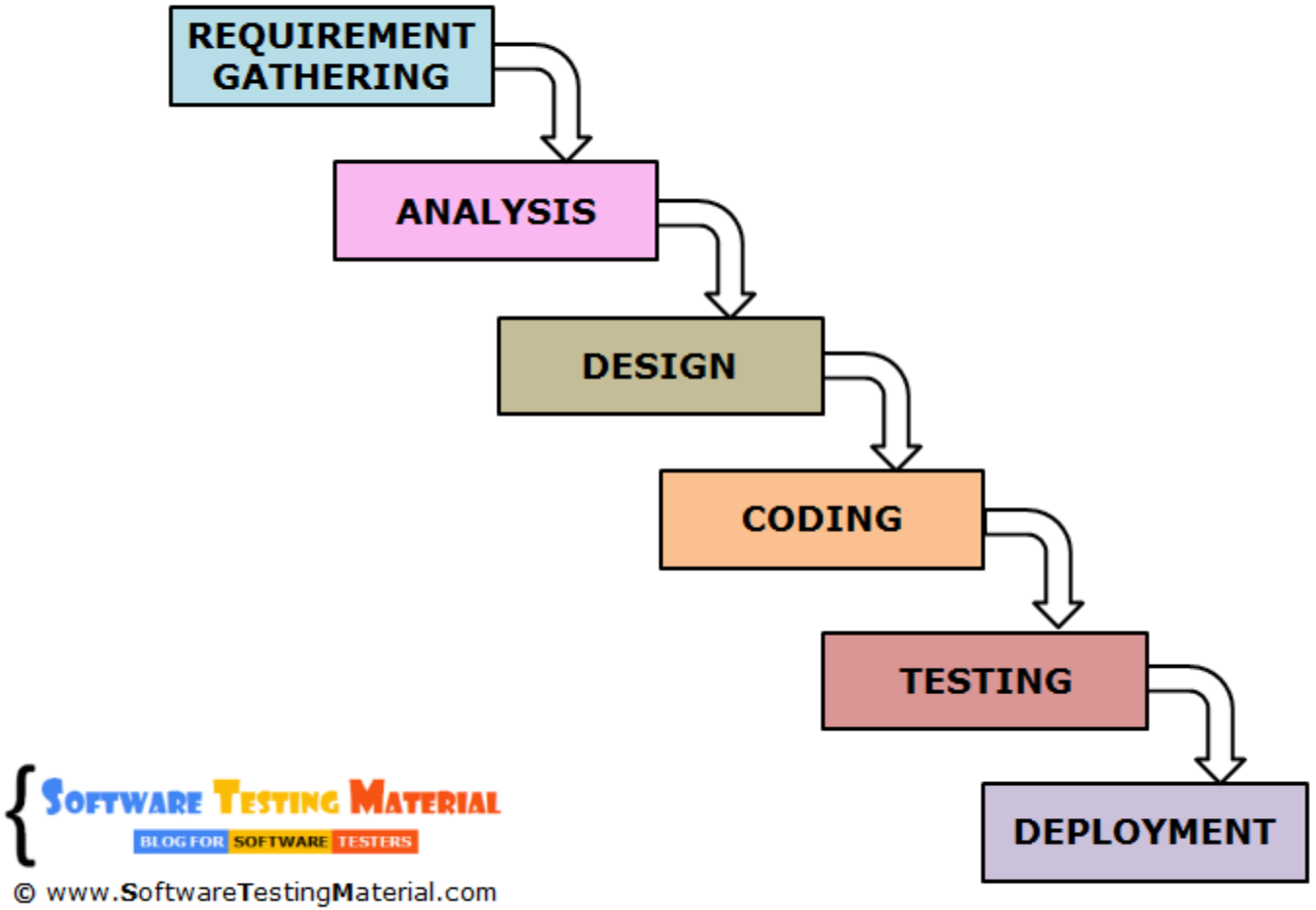
Front End and SDLC



https://arkbauer.com/wp-content/uploads/2019/02/sdlc_software_development_lifecycle.jpg

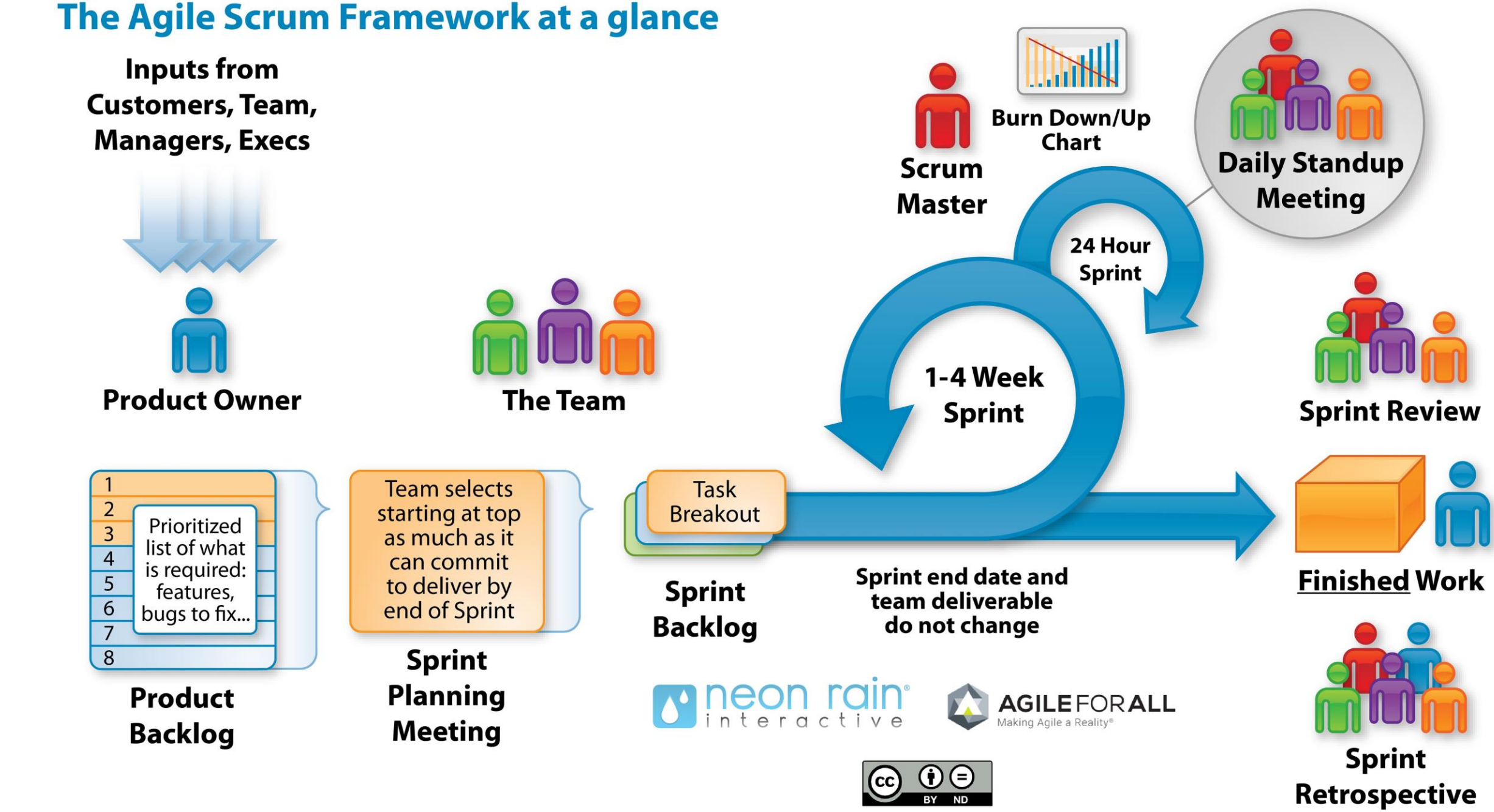
Waterfall Model - SDLC

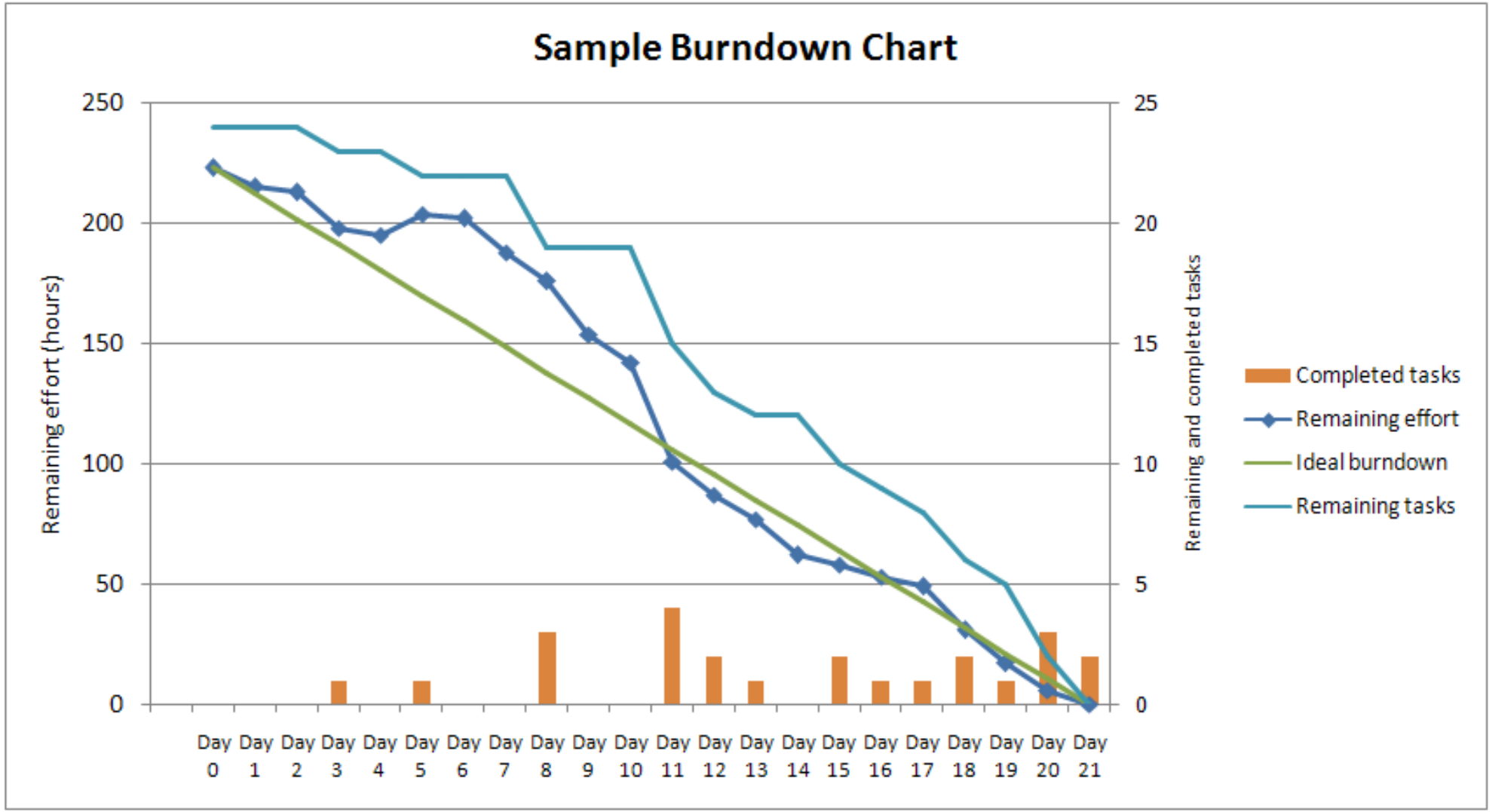


https://external-content.duckduckgo.com/iu/?u=http%3A%2F%2Fwww.softwaretestingmaterial.com%2Fwp-content%2Fuploads%2F2016%2F03%2Fwaterfall-model-sdlc.png&f=1&nofb=1

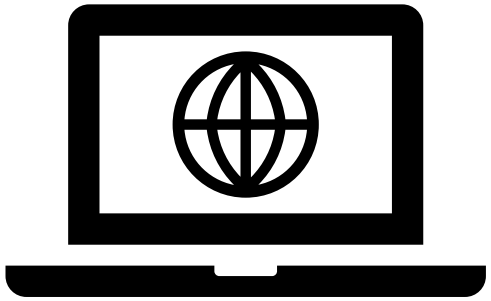
https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fcdn-images-1.medium.com%2Fmax%2F1200%2F0*iDT-e3lTcGR0CSy8.&f=1&nofb=1

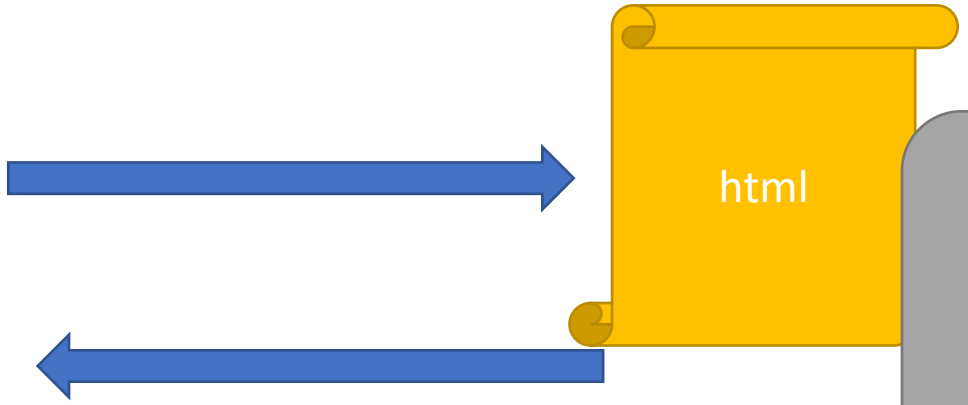
The Agile Scrum Framework at a glance





https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fupload.wikimedia.org%2Fwikipedia%2Fcommons%2F0%2F05%2FSampleBurndownChart.png&f=1&nofb=1

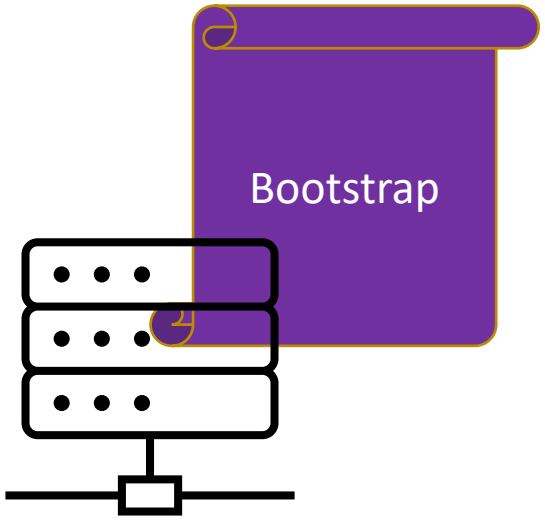




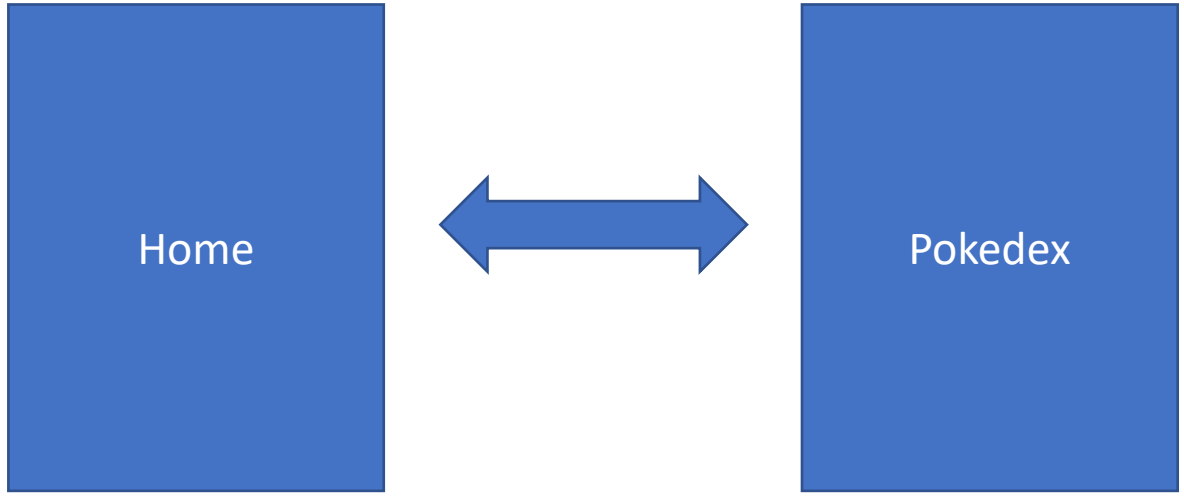
Presentation Layer/ client side/ ui/ux

Data access Layer/ service/ business logic

Data BaseLayer



App design



Local vs Remote Repository

master

Nav-branch

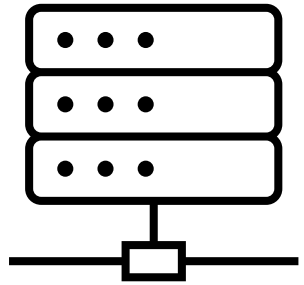
Local repository

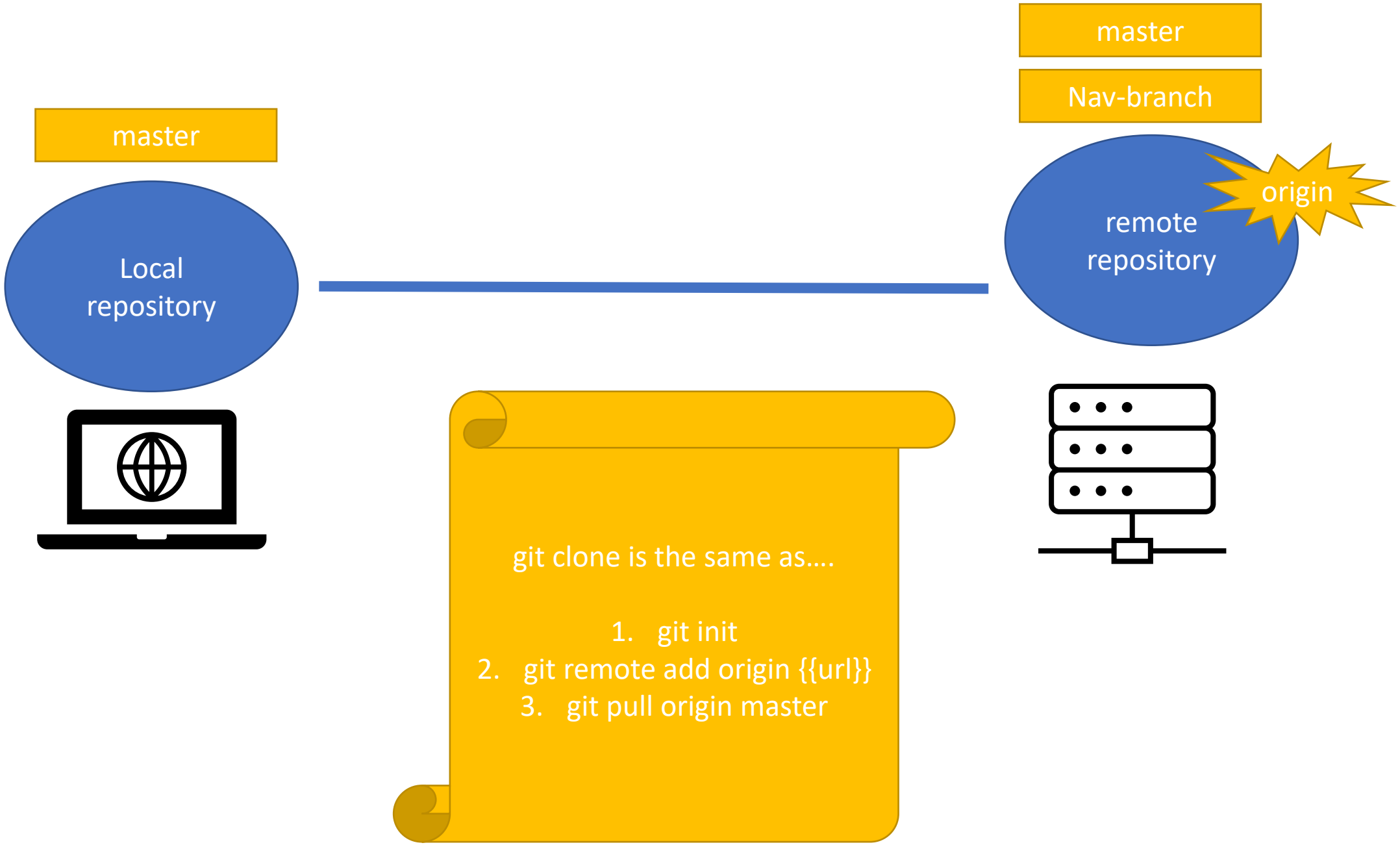
master

Nav-branch

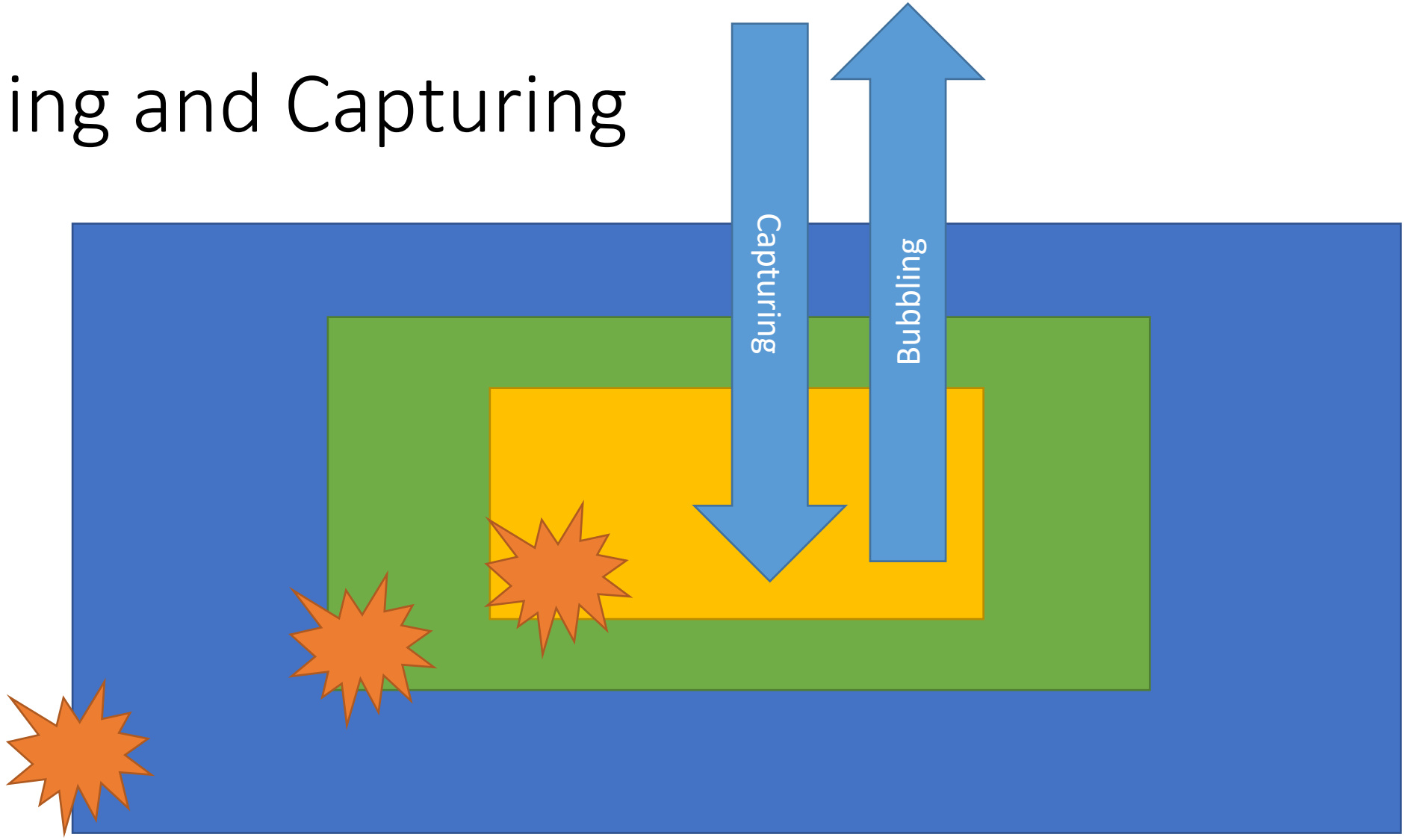
remote repository

origin




master
Local repository
master
Nav-branch
remote repository
origin
git clone is the same as….
1. git init
2. git remote add origin {{url}}
3. git pull origin master

Bubbling and Capturing



Things that I like about the tech.

- How dynamic CSS is
  - We like the Cascading part of CSS -- More consistent
- HTML, CSS, JS … more “simple” (faster to learn)
- We like how modular css and js are
- JS is efficient (less lag)

Things that frustrate me

- Standup meetings (mostly standing up)
- Having to change multiple copies of same html elements in each file
- Having to do paddings and margin for text
  - Changes depending on browser
- Dealing with CSS