

```
1 Education vs training
2
3 TOOLS --
4 Likely to change
5 HR recommendation -- one scripting language and one programming (compiled) language
6 De-emphasize tools -- tool agnostic
7 **Look for what's transferable
8 **Look for the big ideas
9 **Look at the commonalities between like tools
10
11 SPRINT PLANNING AND MANAGEMENT
12 - separate work
13 - assign it/ delegate it
14 - track progress (individual, pod, team, by feature)
15 - track bottlenecks/ dependencies in work
16 - visualization (scrum board, digital scrum boards)
17 - spreadsheet or database-centric systems for tracking hours/ points
18 trello, sure
19
20 VERSION CONTROL
21 - manage changes
22 - collaborate
23 github, git, perforce
24
25 DOCUMENTATION
26 - navigable
27 - helpful
28 - clarity, robustness
29 - cross-linking
30 Confluence, readthedocs
31
32 BUG REPORTING AND FIXING
33 - QA to systematically report bugs
34 - steps for recreation
35 - assign it
36 - test and verify it's fixed
37 web helpdesk, jira, anything with a ticketing system
38
39 CODE REVIEWS
40 - adherence to coding standards
41 - good decisions -- (SOLID), high cohesion, low coupling
42 smartbear, Collaborator, generally these are plugins for version
43
44 TEAM COMMUNICATION
45 - tribal knowledge
46 outlook, discord, slack
```