

1.2

Pros:

- Only have to use and learn one notation
- Easy to compare all models and code, since they are in the same notation.

Cons:

- It can be difficult to represent things like user requirements.
- Developers can easily represent implementation details too early

1.4

Knowledge acquired can be invalidated by new knowledge, as an example most kids believe in Santa Claus and believes that it's through him they get presents at christmas. When they get older they learn that Santa Claus doesn't exist and then all their knowledge of how they get presents at christmas is invalid and they have to learn about it again.

1.6

- The TicketDistributor must enable a traveler to buy weekly passes - functional
- The TicketDistributor must be written in Java - non-functional
- The TicketDistributor must be easy to use - non-functional
- The TicketDistributor must always be available - non-functional
- The TicketDistributor must provide a phone number to call when it fails - non-functional

1.8

- first account: application domain concept
- second account: application domain concept
- third account: solution domain concept (could also be application depends on how you read it).
- fourth account: solution domain concept