Code and Fix

- Informal
- Develop shared understanding, then start coding
- Low overhead

Con:
- Hard to predict time to completion
- Can be at "80%-90% complete" for a long time
- Hard to ensure slow meets quality goals
Waterfall system concept

requirements

high-level design

system architecture

UML

low-level design

implementation:

unit test

integration:

system test

deploy

maintain

slow

very sensitive to requirements errors

document heavy
Evolutionary Prototyping

- Develop a prototype, then iterate on it
  - until customer / developer agree it!
  - good enough

<非执行原型>

<mock-up app>

<执行原型>

- fast-to-create
- not so great on investment in final product
- easier to change
- can do walk-throughs with users - early
- easy to reason about
Jin's Web App Welcome

Login

username

password