UNIVERSITY of WASHINGTON

CSE 583

Software Engineering for Data Scientists
Working in Teams

David Beck, Joseph Hellerstein, Jake VanderPlas, Alex Ford

The University of Washington

Oct 27, 2017

UNIVERSITY of WASHINGTON
eScience Institute

Agenda

- Software licenses
- Software development overview
- Team process with in-class exercises:
  - Code reviews
  - Technology review
  - Standups
  - Team Building
- Team formation

Software Licenses

Overview of Software Licenses*

A software license is a legal instrument (usually by way of contract law, with or without printed material) governing the use or redistribution of software. Under United States copyright law all software is copyright protected, in source code as also object code form. The only exception is software in the public domain. A typical software license grants the licensee, typically an end-user, permission to use one or more copies of software in ways where such a use would otherwise potentially constitute copyright infringement of the software owner's exclusive rights under copyright law.

*Mark Webbink

Software Licenses*

Rights granted	Public domain	Non-protective FOSS license (e.g. BSD license)	Protective FOSS license (e.g. GPL)
Copyright retained	No	Yes	Yes
Right to perform	Yes	Yes	Yes
Right to display	Yes	Yes	Yes
Right to copy	Yes	Yes	Yes
Right to modify	Yes	Yes	Yes
Right to distribute	Yes	Yes, under same license	Yes, under same license
Right to sublicense	Yes	Yes	No
Example software	SQLite, ImageJ	Apache Webserver, ToyBox	Linux kernel, GIMP

FOSS = Free and Open Source Software

*Mark Webbink

Software Licenses*

Rights granted	Freeware/Shareware/ Freemium	Proprietary license	Trade secret
Copyright retained	Yes	Yes	Yes
Right to perform	Yes	Yes	No
Right to display	Yes	Yes	No
Right to copy	Often	No	No
Right to modify	No	No	No
Right to distribute	Often	No	No
Right to sublicense	No	No	No
Example software	Irfanview, Winamp	Windows, Half-Life 2	Server-side World of Warcraft

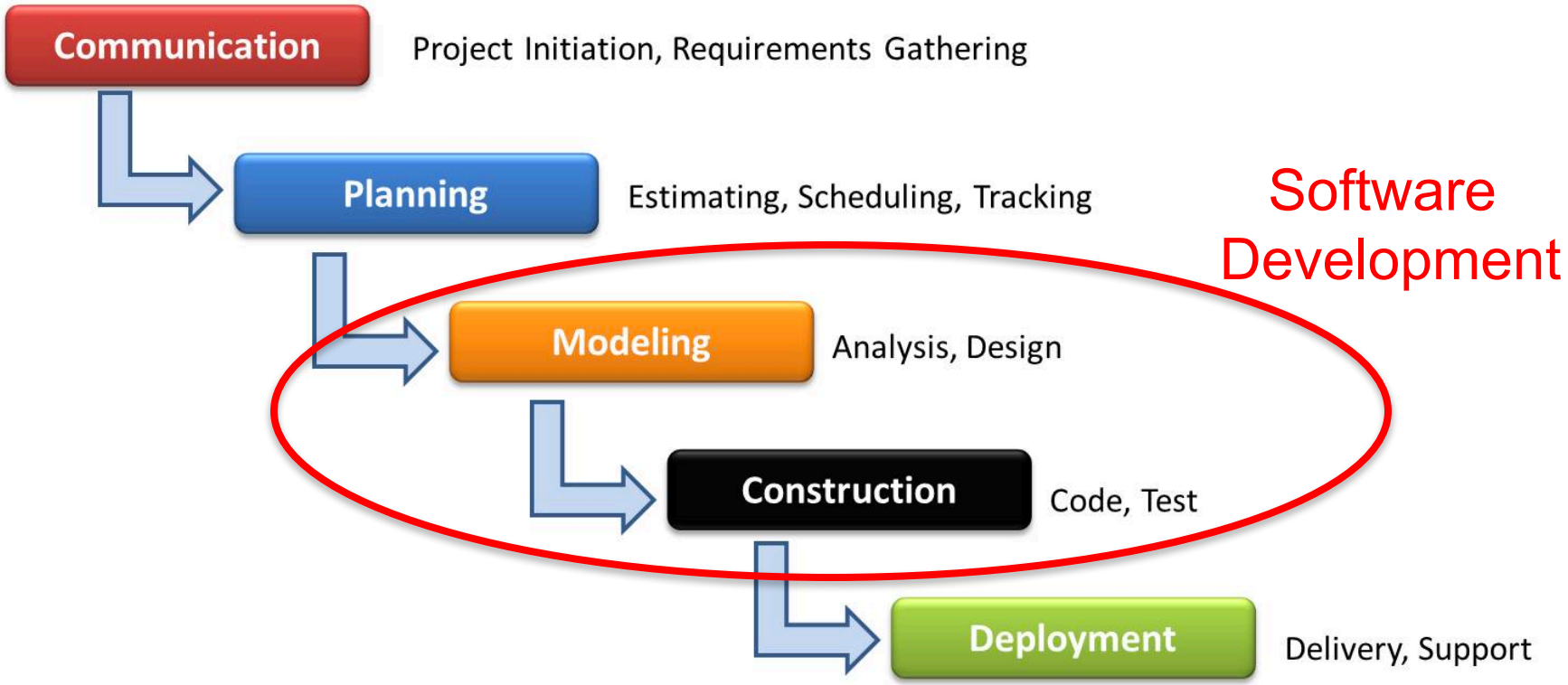
The default (no license): No rights are granted.

*Mark Webbink

Software Development Overview

Software Development Phases

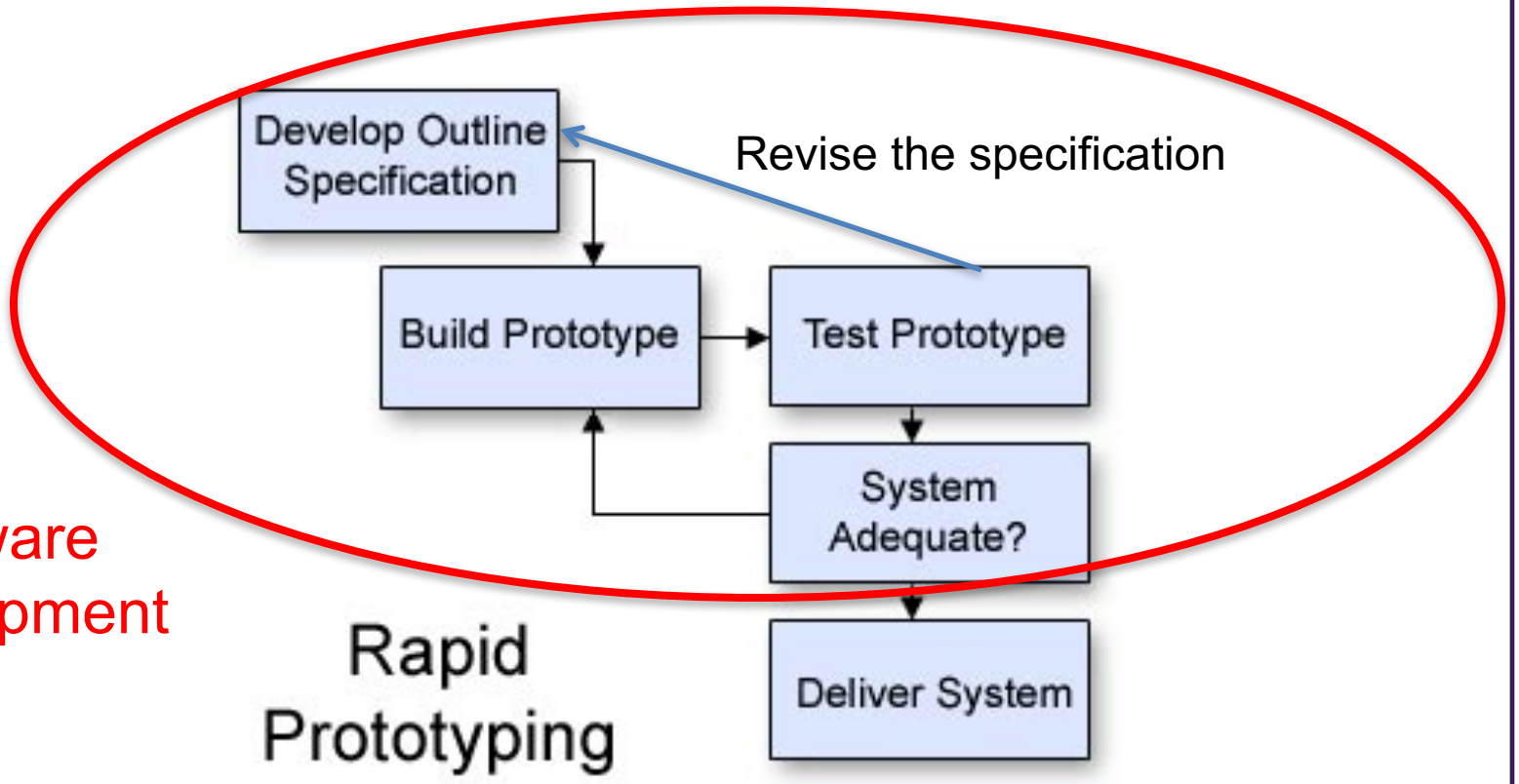
Waterfall Process Model



Why does this work poorly?

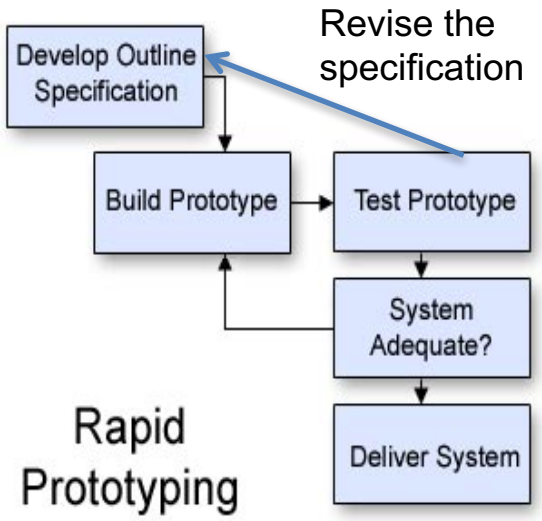
Rapid Prototyping

- Why?
  - Cannot specify all requirements in advance



Team Process

Team Activities



- Reqs gathering (functional spec.)
- Design
  - Technology assessments
  - Write specifications
  - Review specification
- Implementation
  - Code
  - Code review
- Bug prioritization and resolution
- Standups (status update)

Projects

Project Updates

- What is your data?
  - You should have 2 datasets in hand!
- Who are your users?
  - General public? Scientists? Analysts?
- What questions are users trying to answer?
- What are the use cases (user-system interactions) to answer their questions?
- What issues are there ("known unknowns") with building your system?

Code Review

Code Review Template

- Why code review?
  - Improve code quality and find bugs

- Background
  - Describe what the application does
  - Describe the role of the code being reviewed
- Comment on
  - Choice of variable and function names
  - Readability of the code
  - How improve reuse and efficiency
  - How use existing python packages

Code & Test

```
'''
   Code to find prime numbers
'''

# Prime number finder with a logic bug
def check_prime(num):
  """
  Checks to see if a number is prime.
  :param int num:
  :returns bool:
  """
  is_prime = True
  for i in range(1,num):
    if (num % i) == 0:
      is_prime = False
      break
  else:
    pass
  return is_prime

# Run the following code if the file is run at the command line
if __name__ == "__main__":
  num = int(input("Enter a number: "))
  if primeChecker(num):
    print ("Is prime!")
  else:
    print ("Not a prime.")
```

```
import unittest
from prime import check_prime

# Define a class in which the tests will run
class PrimeTest(unittest.TestCase):

    def test_smoke(self):
        check_prime(3)

    def testSimple(self):
        self.assertTrue(check_prime(3))


if __name__ == '__main__':
    unittest.main()
```

- Background
  - Describe what the application does
  - Describe the role of the code being reviewed
- Comment on
  - Choice of variable and function names
  - Readability of the code
  - How improve reuse and efficiency
  - How use existing python packages

Technology Review

Technology Review Template

- Why technology reviews?
  - Determine if use a package

- Background
  - Requirements that indicate a need for the proposed package
- Discuss
  - How the package works
  - Appeal of using the package
  - Drawbacks of using the package

Example of A Technology Review

Antimony Package for Kinetics Modeling

- Background
  - Need kinetics models to explore certain what-if questions in chemical systems.

Using Antimony

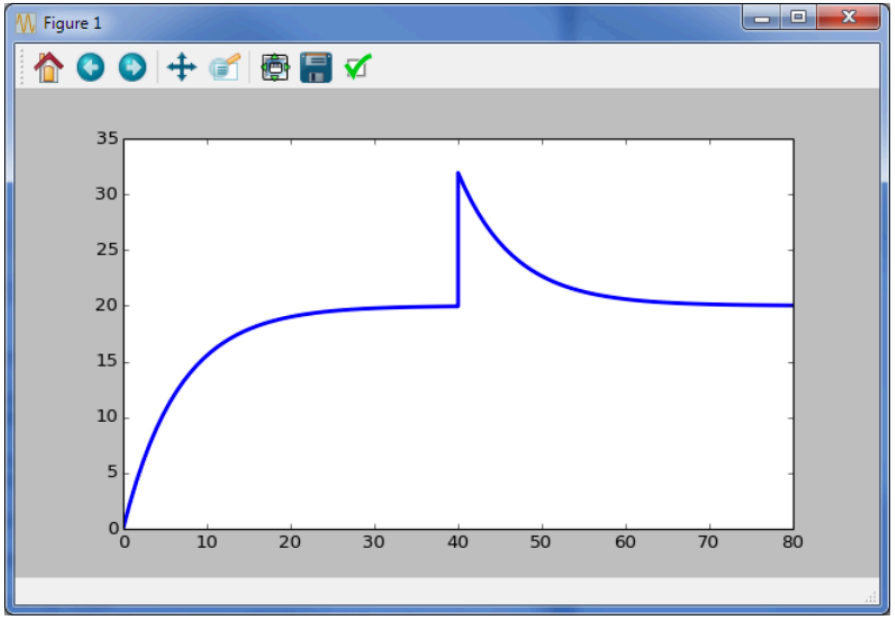
```
import numpy # Required for vstack
import tellurium as te

rr = te.loada ('''
     $Xo -> S1;  k1*Xo;
      S1 -> $X1; k2*S1;

     Xo = 10; k1 = 0.3; k2 = 0.15;
''')

m1 = rr.simulate (0, 40, 50)
```

Kinetics model is a python string



Perturbation of *S1*

Assessment of Antimony

- Appeal
  - Readable kinetics models
  - Can use python with Antimony
  - Exports to and imports from SBML (systems biology modeling language)
- Drawbacks
  - Poor support for the package
  - Scaling may be a problem

Standups

Standup Template

- Why standups?
  - Communicate status and actions within and between teams
- Should be presented in 1-2 minutes

- Progress this period
  - How it compares with the plan
  - If behind plan, how compensate to make plan end date
- Deliverables for next period
- Challenges to making next deliverables such as:
  - Technology uncertainties and blockers
  - Team issues

Team Building (Team Size: ~4)

Forming A Team

- Have a clear problem statement. Examples
  - Lab automation for bioreactors
  - Predict housing prices in King County
- Have data that relates to the problem
- Find others who are interested in your problem
- Team commitment. Agree on:
  - Questions to answer
  - Have data that can answer the questions

Team Formation

Structure of Breakout

- Give a short statement of your interests and the data
- Talk, talk, talk
- Form team. Send team name and members to Alex.
- HW4 is mostly a team homework

Teams

1. Anna – IMDB, Movie DB – estimate movie income (5).
2. Patrick – Reading score data (3).
3. Oriana – Climate and stream flow data and agriculture data (5).
4. Tian Qi – Real time data car accidents (MS) & traffic data from (Google Maps) for route planning for police response (5).
5. Ryan – Utility and weather to maximize cost savings; also, Google sunroof (by zip code; position on the roof for the solar installation) (5).
6. Pranay – Social network data (4).
7. David – Find membrane proteins and characterize them (3).
8. Ty – Transportation & social science data (4).
9. Tejas – Credit card fraud detection. 1 dataset and looking for second. (4)