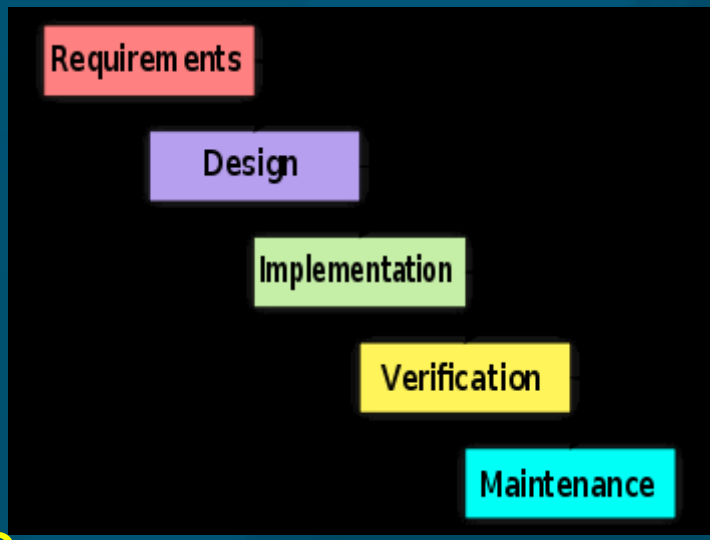
Software Development Models
Doctest Module

4 Dec 2009
CMPT140
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Top-down development

- WADES!
- Waterfall model: assumes every step is done perfectly
- But in the real world:
  - Written requirements change (client changes mind)
  - Apprehension of the requirements is fuzzy
  - Design is incomplete at first
  - Execution is sloppy (“spaghetti code”)
  - Scrutinization results in endless debugging!



Software development process

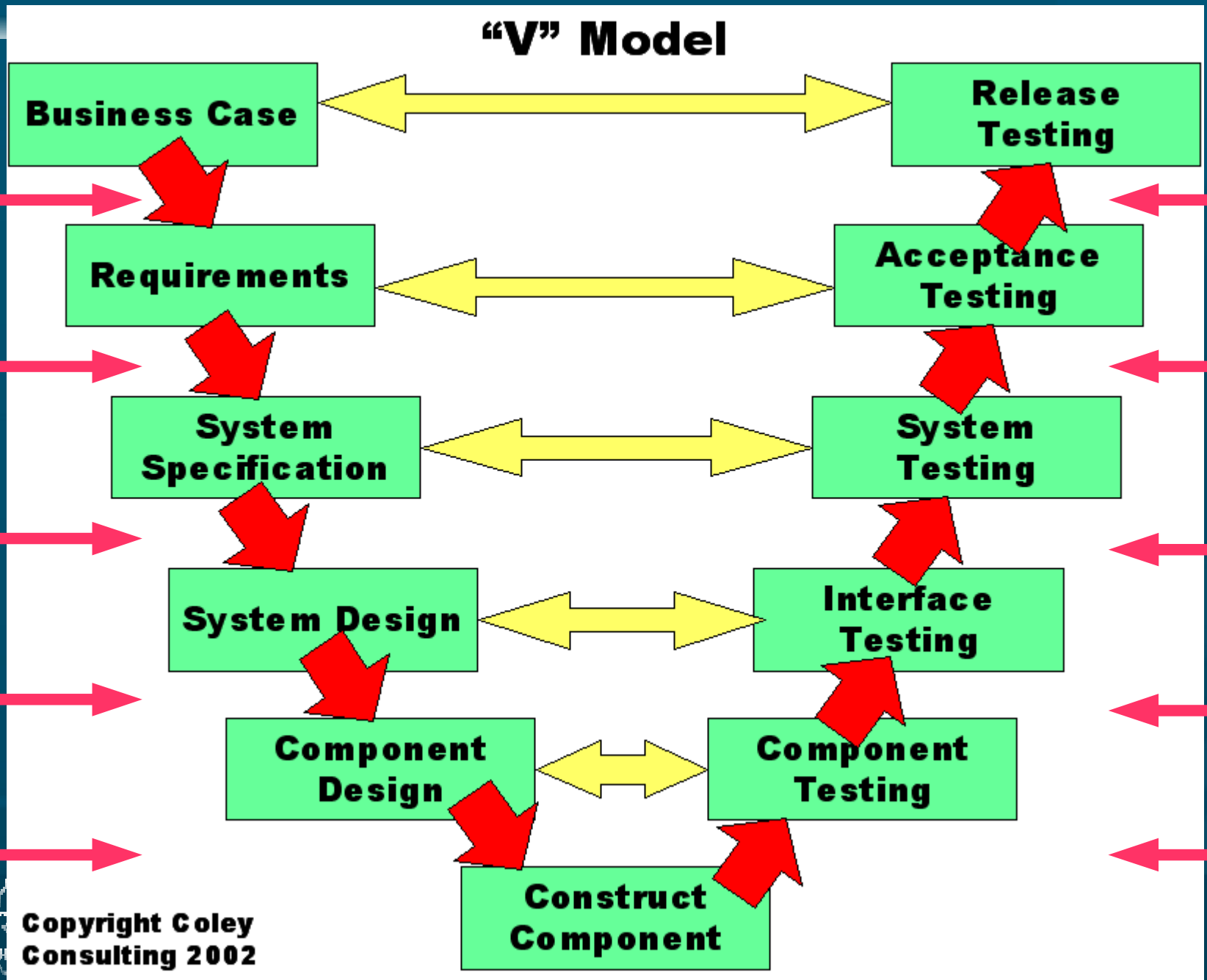
- Lots of people have tried to design better ways to develop that reflect the real world:
  - Development process: how you do the work
- No silver bullet: different projects, different people may require different processes
- Be flexible: your future employer may demand that you use a particular process
  - Or might not have any process in mind: then it's up to you to structure your time!
  - Get results; make the client happy!

V development model

- Design from Top-Down:
  - What does the client want? (requirements)
  - What will our system do? (specification)
  - How will we do it? (design)
  - What components will we need?
- Test from Bottom-Up:
  - Does each component work as it should?
  - Do the components integrate correctly?
  - Does it do what we promised it would?
  - Is the client happy?

V model (Coley)

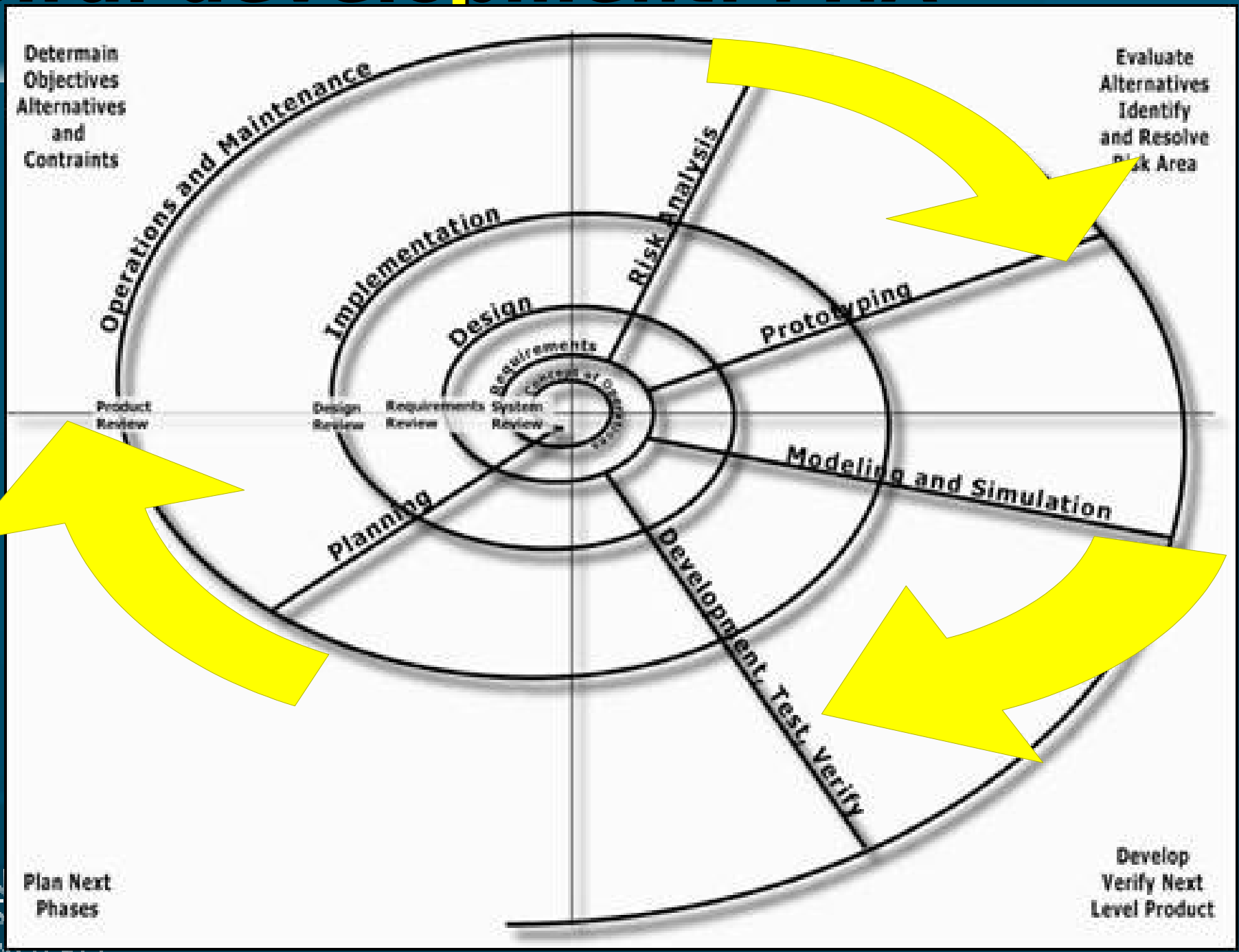
gate



Spiral development model

- The spiral model is an iterative waterfall:
  - Repeat parts of WADES, refining as you go
- Get a prototype out early
- Get feedback early
  - Don't waste time developing what the client doesn't want
- Client: “I'll know it when I see it”
  - Developer: “Is this what you want?”
- Anticipate several cycles/refinements!

Spiral development: FHA



XP: Extreme Programming

- Extreme programming was coined by Kent Beck in 1999 while developing Chrysler's payroll sys.
- Form of spiral, with more and tighter spirals:
  - Code: if there are two different solutions, implement both!
    See which works better.
  - Test: it's the only way to be sure it works
- Values: Communication, Simplicity, Feedback, Courage, and Respect

Agile development

- XP is an example of agile development:
  - Get results quickly
  - Adapt to changing requirements
- "Agile Manifesto" philosophy:
  - Individuals+interactions vs. processes+tools
  - Working software vs. comprehensive docs
  - Customer collaboration vs. contract negotiation
  - Responding to change vs. following a plan

Agile and the Toyota Way

- Agile does not mean total anarchy!
  - Clear goals, communication with client
  - Rapid development with frequent feedback
- Agile influenced the "Toyota Way":
  - Long-term philosophy/goal
  - The right process
  - Invest in people
  - Continuously solve root problems

Test-first development

- Frequent and thorough testing is an important component of all these development models!
- Test-first development:
  - Write your test cases before you code!
  - Once you have the design/specification (e.g., pre/post-conditions of a method),
  - Write test cases in the comments/docstring
  - Then code and test along the way
- Python's doctest module makes this easy

doctest: Python test cases

- In your docstrings, include test cases as:
  - **>>> run_my_method(5)**
  - **“expected output”**
- Prefix commands with '>>>'
  - Can have multiple commands (script)
- Put expected output on line by itself
  - If you expect an exception, write the expected “red text” for the exception!
- See factorialtest.py example
- More details in Python doctest library docs