

---

# Testing in the Lifecycle

Ajitha Rajan



## Tasks to be completed

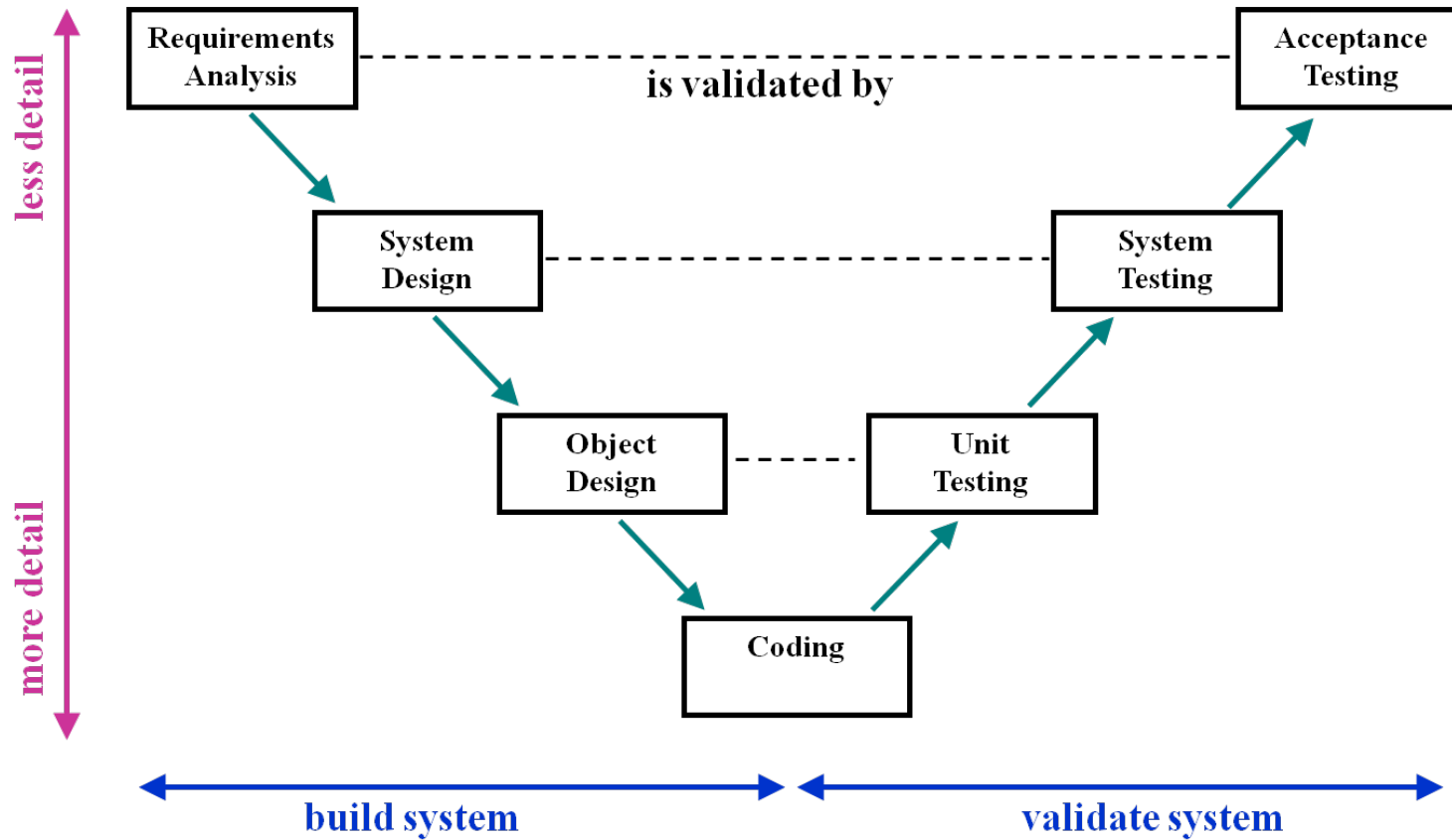
1. Form groups of 3 or 4 and email the TA with
  - **Names** and UUN of group members
2. Sign up on Piazza discussion forum. You will also receive an email about it.

## Recap: Waterfall model of software development

1. Requirements
2. Design
3. Implementation
4. Testing
5. Release and maintenance

*Sequential, no feedback* — Ironically its “author”, Royce, presented it as an example of a broken model

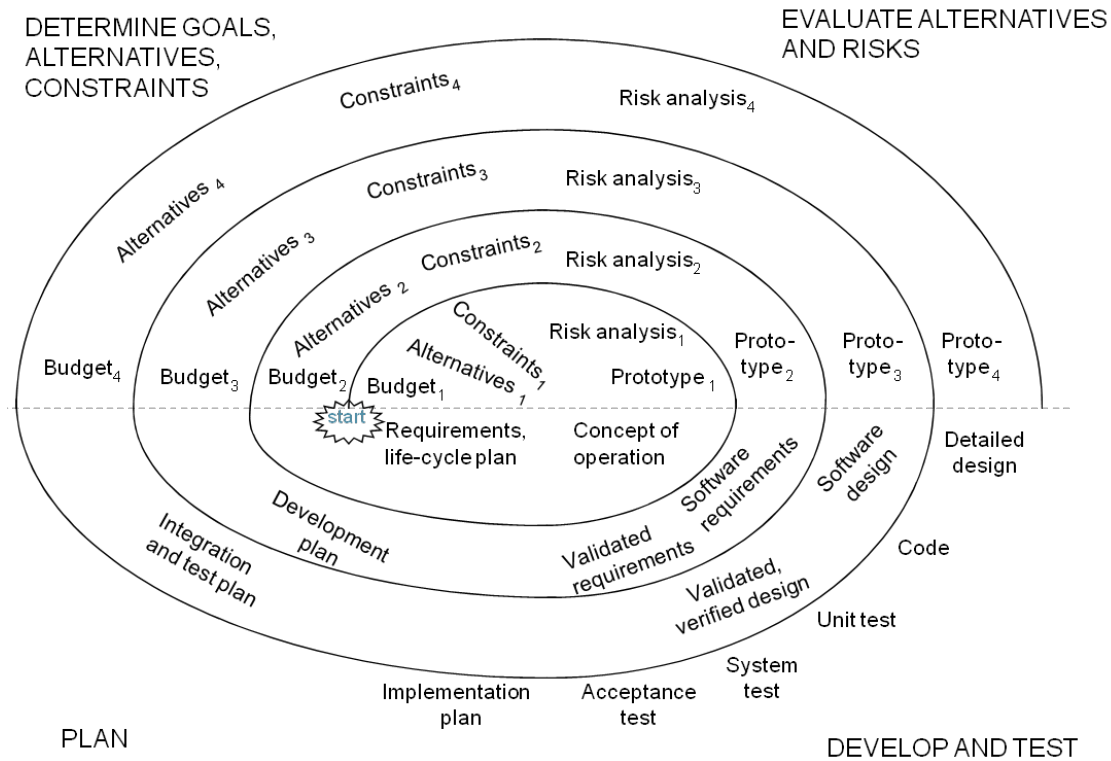
# V-model



## V-model Rationale

- This is a modified version of the waterfall model.
- Tests are created at the point the activity they validate is being carried out. So, for example, the acceptance test is created when the systems analysis is carried out.
- Failure to meet the test requires a further iteration beginning with the activity that has failed the validation.
- V-model is focused on creating tests in a structured manner.
- It is popular with developers of systems that are highly regulated because it is well suited to creating evidence that can be used to justify a system to a regulator.

# Boehm's Spiral Model



## Spiral Model Rationale

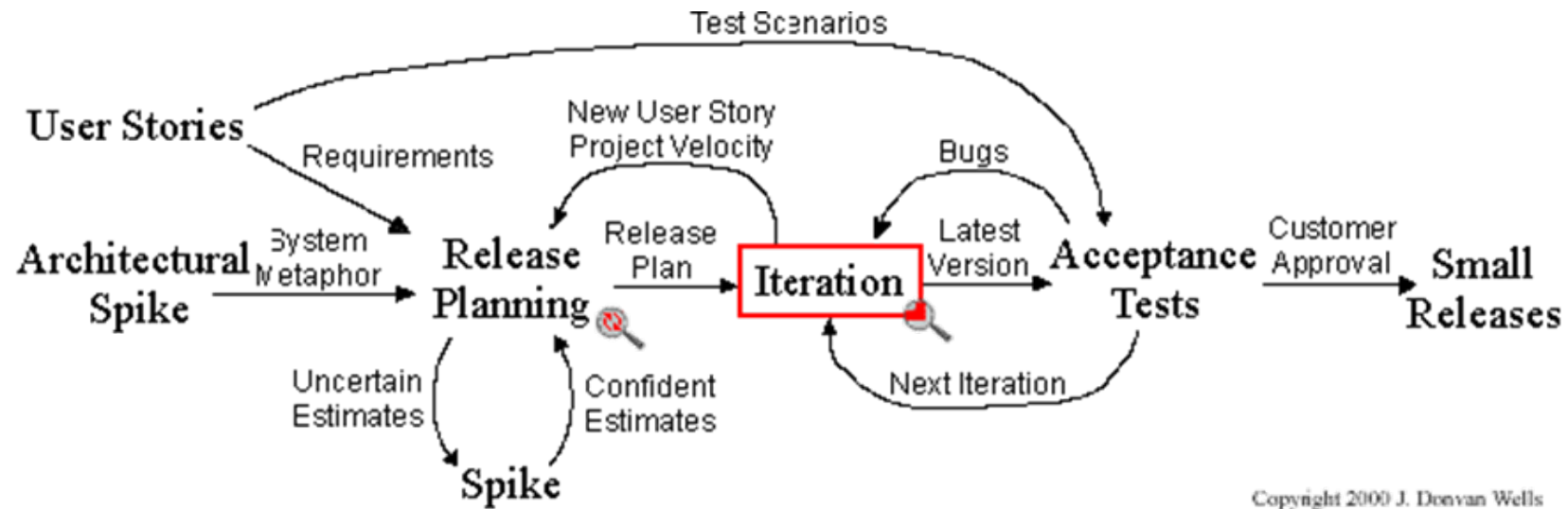
- The spiral model is focused on controlling project risk and attempting formally to address project risk throughout the lifecycle.
- V&V activity is spread through the lifecycle with more explicit validation of the preliminary specification and the early stages of design. The goal here is to subject the early stages of design to V&V activity.
- At the early stages there may be no code available so we are working with models of the system and environment and verifying that the model exhibits the required behaviours.

## XP principles

- eXtreme Programming advocates working directly with code almost all the time.
  - The 12 principles of XP summarise the approach.
  - Development is test-driven.
  - Tests play a central role in refactoring activity.
  - “Agile” development mantra: Embrace Change.
1. Test-driven development
  2. The planning game
  3. On-site customer
  4. Pair programming
  5. Continuous integration
  6. Refactoring
  7. Small releases
  8. Simple design
  9. System metaphor
  10. Collective code ownership
  11. Coding standards
  12. 40-hour work week



# eXtreme Programming (XP)



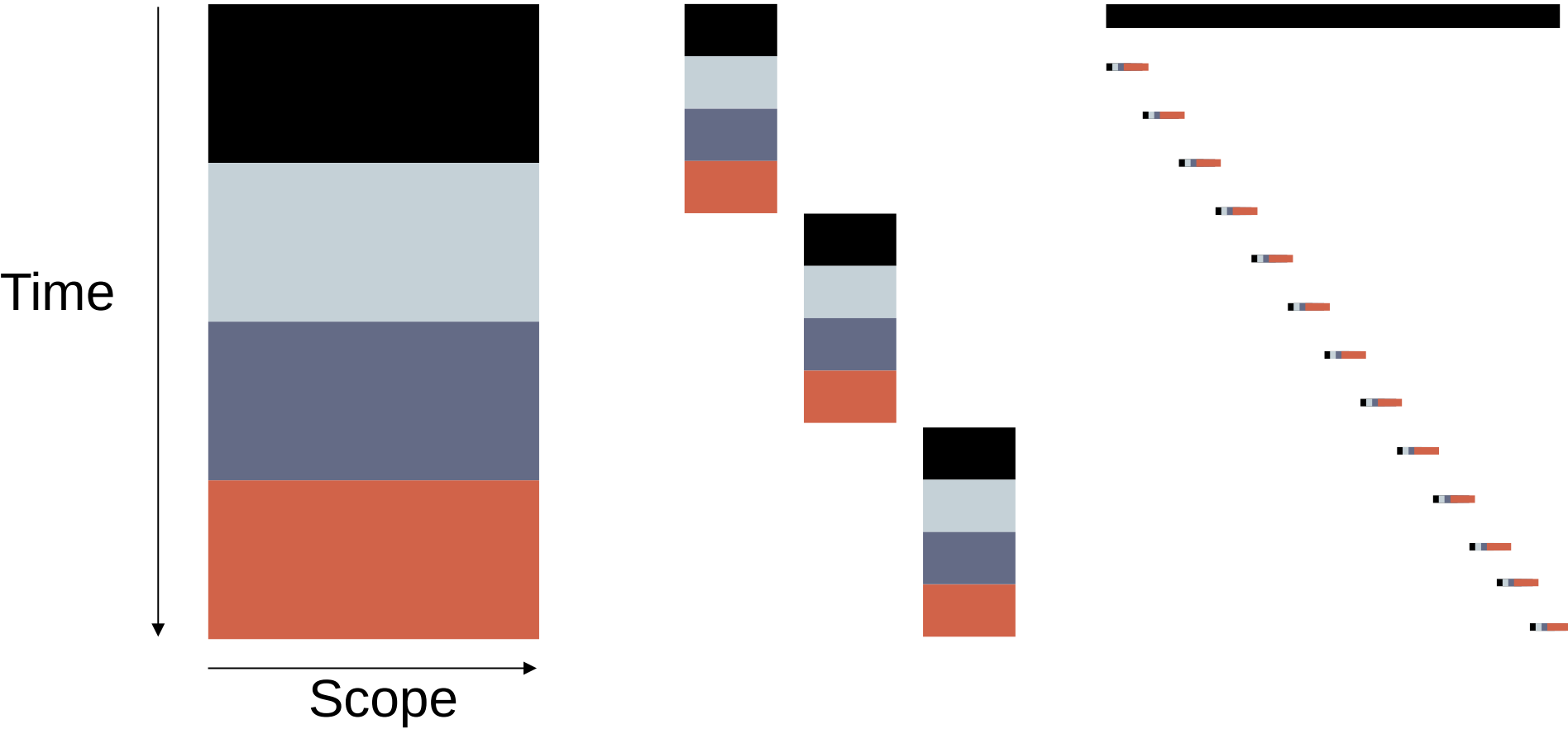
<http://www.extremeprogramming.org/map/project.html>

# Common Software Development Processes

Waterfall

Iterative

XP



Slide adopted from Beck

# Facebook's Process Model

“**Perpetual development-** As many internet companies Facebook adopts a *continuous development model*. In this model, software will never be considered a finished product. Instead features are continuously added and adapted and shipped to users. *Fast iteration* is considered to support rapid innovation. For its web version, Facebook pushes new changes in the code twice a day.

“It would be impossible to foresee a-priori how a new feature would be used by the hundreds of million of people using Facebook services every day. The different uses influence the way a new feature is shaped and further developed.”



“Moving Fast with Software Verification”