Comp8081
Management Issues in Software Engineering

Donna Turner

Agenda

- Attendance
- Review Core Issues in Rapid Development (McConnell – Chapter 6)
- Lifecycle Planning (McConnell – Chapter 7)
- Summary
- For next week

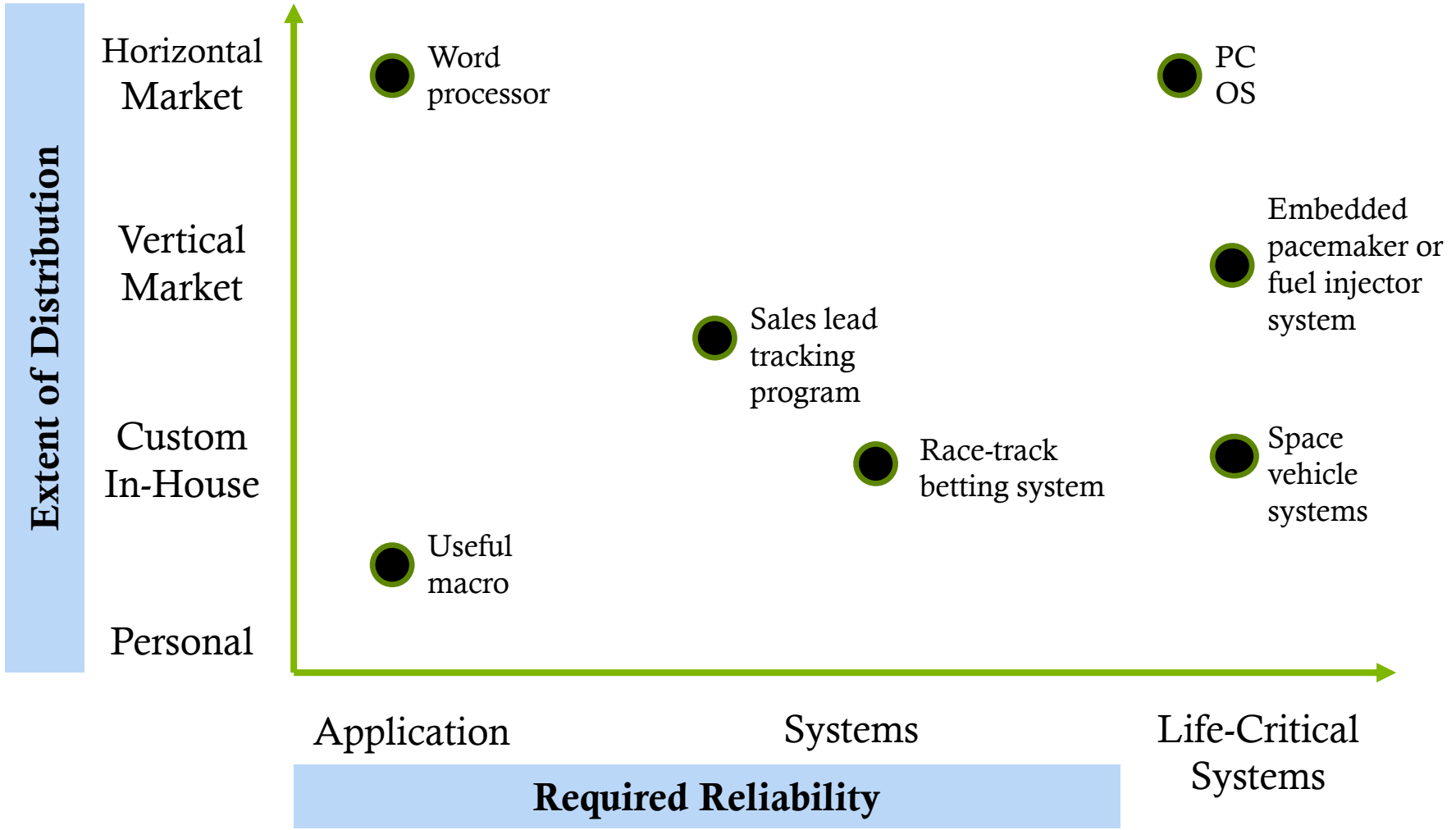
Review

Core Issues in Rapid Development
- McConnell, Chapter 6

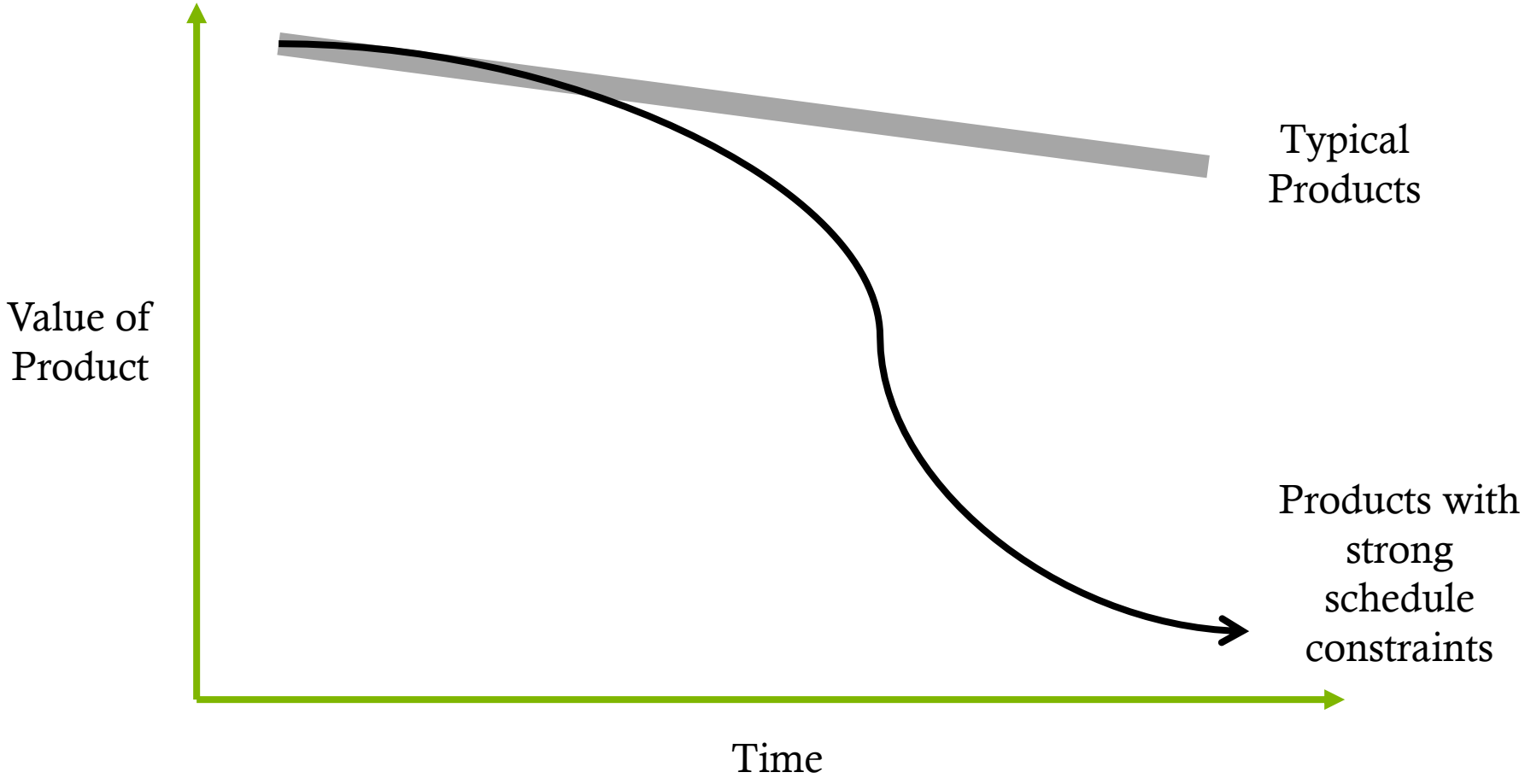
One size fits all?

Distribution and Reliability

Adapted from McConnell pg. 110

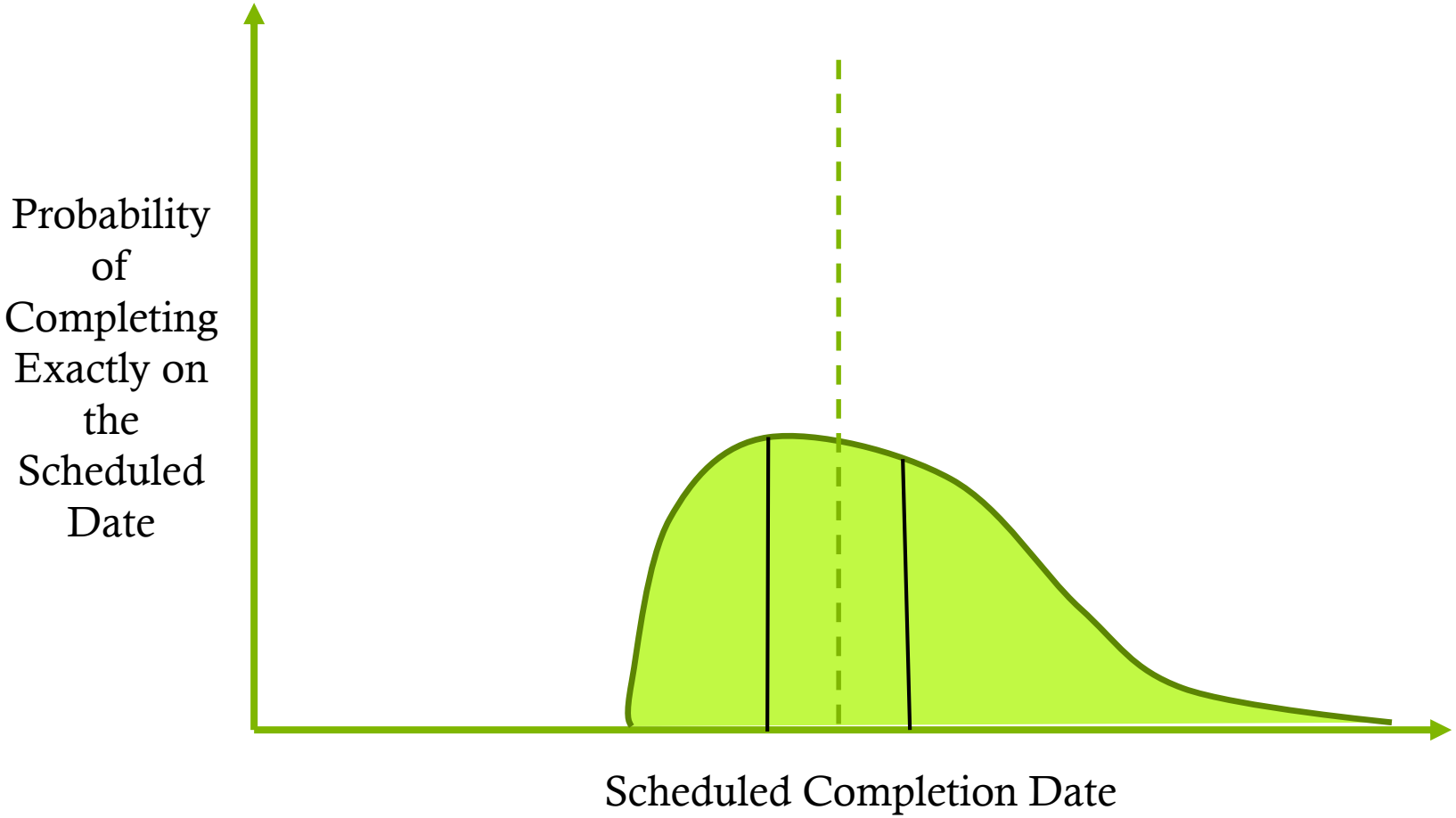


Value over Time



Adapted from McConnell pg. 112

Odds of Completing on Time



Adapted from McConnell pg. 117-118

Rapid Development Look-Alikes

- Runaway Prevention
- Predictability
- Lowest Cost
- Fixed Drop-Dead Date
- Unpaid Overtime

Questions to ask

On D2L, look at Lifecycle Planning, list of projects

- How widely **distributed** will the product/system be?
- How importance is **reliability** of the product/system?
- Is this a case of a "look-alike" project?
- How "carefully" does the product need to be developed?
  - Is there a drop-dead date that you need to meet? Really?
  - Can you afford to increase risk by reducing the schedule?

(Class discussion afterwards)

Life Cycle Planning

McConnell, Chapter 7

Lifecycle Planning

- *Lifecycles*™ from Life Fitness™ are very important for developers
  - This is a sedentary industry
  - Sitting is the new smoking
- Business goals are:
  - Productivity
  - Creativity
  - Absenteeism
  - Deaths

Lifecycle Planning

Some questions to start

- From the Core Issues review, which factors can affect life cycle model choice?
- How do we ensure the goals are met, within the constraints we have, while managing risks appropriately?
- Is there a one-size-fits-all approach?
- Which lifecycle models do you have experience with?

Lifecycle – Planning – Model

Lifecycle

Applies to every software dev effort

All activities between "Idea" to "End of Life"

Lifecycle Planning

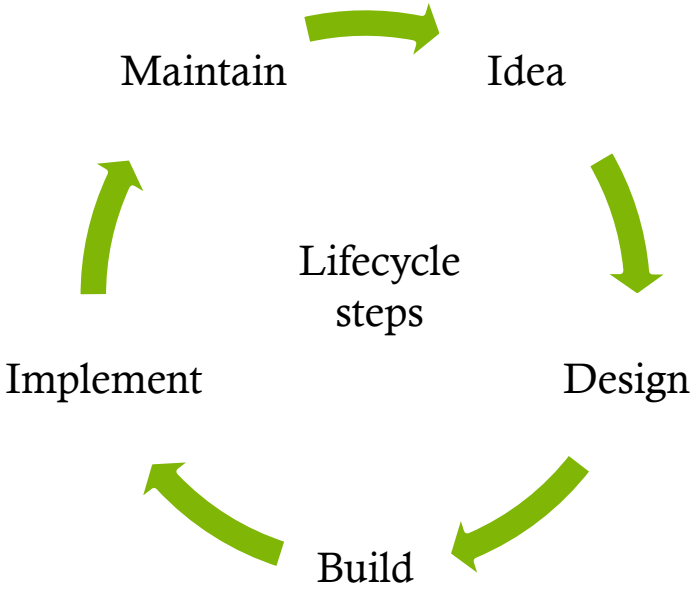
Balancing various considerations to determine which Lifecycle Model to select

Lifecycle Model

Prescriptive model of events, order and what should happen in between dev events

Lifecycle

- Applies to every software dev effort
- McConnel says: all activities between “Idea” to “End of Life”
- PMI says: a project has a defined beginning and end, and is unique
  - So all activities from “Idea” to “first delivery/installation”



Lifecycle Planning

Different projects have different development needs –
even if they all need to be developed as soon as possible

Requirements Considerations

How well do my customer and I *both* understand the requirements at the beginning of the project?

How likely are requirements to change?

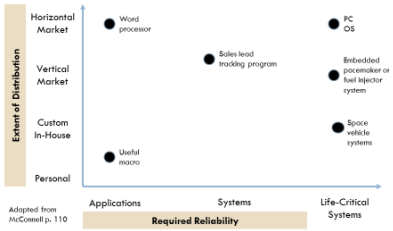
Architectural Considerations

How well do I understand the system architecture?

Am I likely to need to make major architectural changes midway through the project?

Reliability Considerations

How much reliability do I need?



Future Version Considerations

How much do I need to plan ahead and design ahead during this project for future versions?

Lifecycle Models

- Prescriptive model of what should happen in between those events
- Establishes order in which a project specifies, prototypes, designs, implements, reviews, tests and performs other activities
- To select a model, you need **Lifecycle Planning**

Beware "The Straw Man"

- A straw man is a common type of argument and is an informal fallacy based on the misrepresentation of an opponent's argument
- To be successful, a straw man argument requires that the audience be ignorant or uninformed of the original argument

From Wikipedia

If you're going to debate or compare, base it on facts not assumptions

Straw Man examples

- Person A:
  We should relax the laws on beer
- Person B:
  No, any society with unrestricted access to intoxicants loses its work ethic and goes only for immediate gratification.

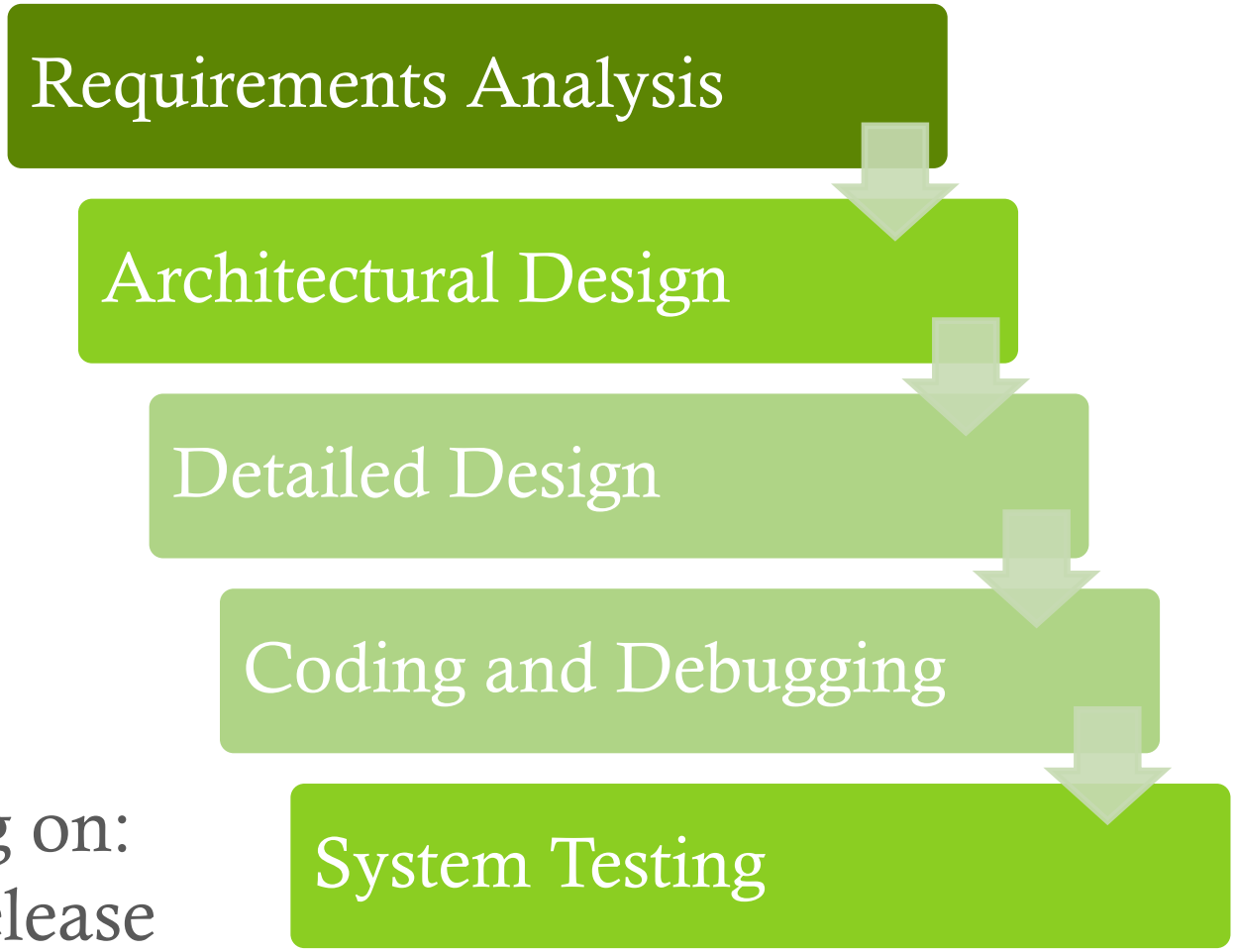
- Richard Nixon 1952 "Checkers Speech"
  - Accused of illegally appropriated $18,000 in campaign funds for personal use
  - Spoke about a gift from a supporter, a dog named Checkers, that his family had decided to keep

Why Mention This?
Often we will disregard different Lifecycle Models without actually knowing the facts or based on pre-conceived notions.

Case Study 7-1

- Read the Case Study on page 134
  - What life cycle was chosen?
  - How was it chosen?
  - Was LifeCycle Planning properly done?

Pure Waterfall Process



We are focusing on:
idea to initial release

Waterfall Process

Characteristics

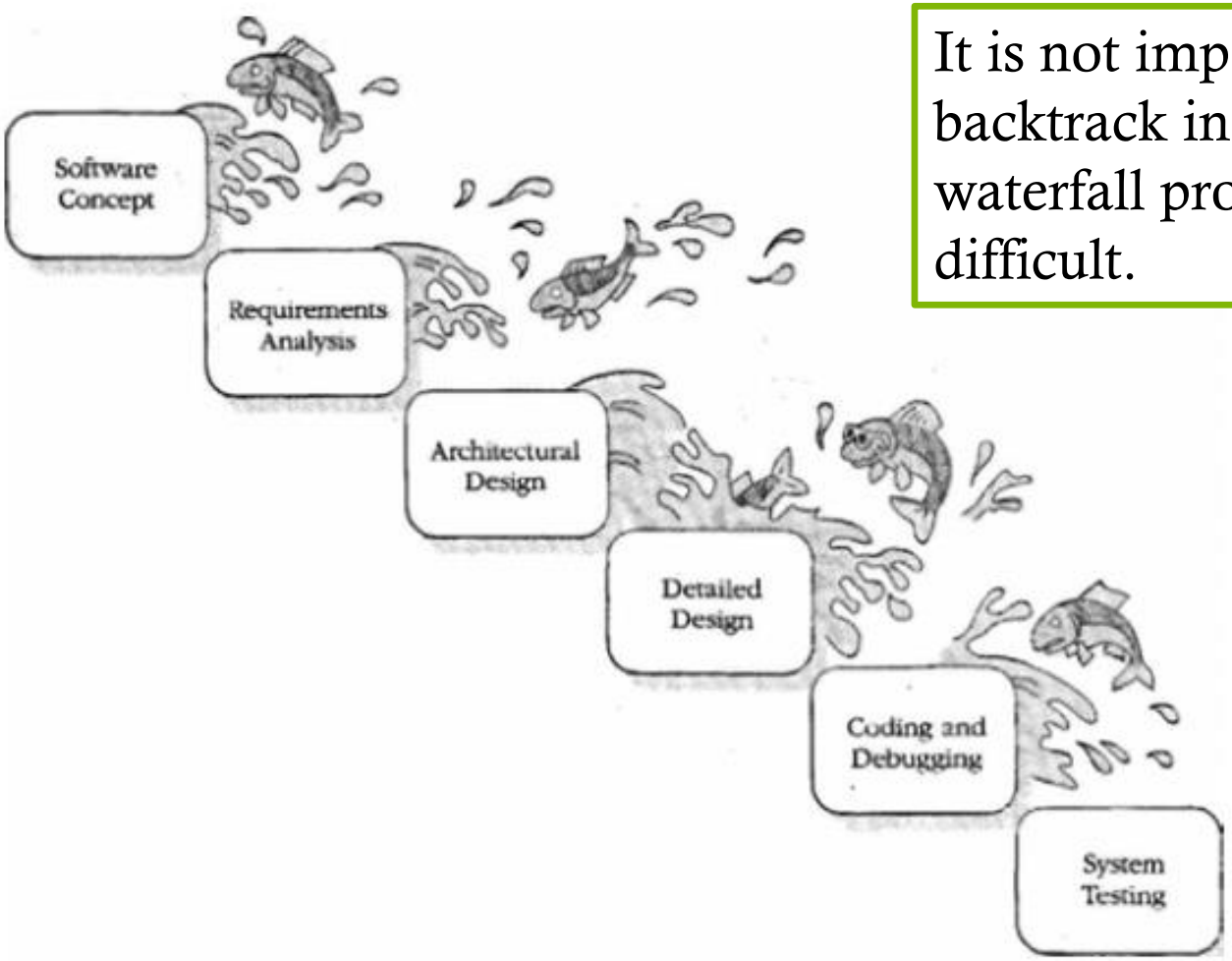
- Documentation and planning driven
- Discrete, contiguous stages
- Front-end loaded activities

Benefits and Drawbacks

- Benefits
  - Minimizes planning overhead, enables requirements stability
- Drawbacks
  - Requires early, full requirements specs
  - Less flexible

We can compare to and contrast with the *actual* Waterfall process
Please don't succumb to the straw man fallacy
e.g. Waterfall is old so it is bad, or Waterfall has too much documentation

Waterfall process



It is not impossible to backtrack in a pure waterfall project, just difficult.

Figure 7-2. *Another depiction of the waterfall model—the salmon lifecycle model, It inn 't impossible to back up using the waterfall model, just difficult.*

More Lifecycle Models

- Some to Explore
  - Spiral
  - Modified Waterfall – 3x
  - Evolutionary Prototyping
  - Design-to-Schedule
  - Staged Delivery
  - Evolutionary Delivery, including Incremental Development Practices
- Some others
  - Code-and-Fix
  - Design-to-Tools
  - COTS
  - Scrum
  - XP

Lifecycle Planning Discussion

Groups

We will divide up the Lifecycle models

Review and summarize

Then in-class debrief

- The elevator pitch:
  - What is the summary?
  - Which risks does it aim to manage?
  - When should it be applied – what's the value?

Model	Reference
Spiral	7.3 (141)
Modified Waterfall – 3x	7.4 (143)
Evolutionary Prototyping	7.5 (147)
Design-to-Schedule	7.7 (149)
Staged Delivery	7.6 (148)
Evolutionary Delivery, including Incremental Development Practices	7.8 (151)

Discussion Questions

- Could you "Mix 'n' Match" these approaches?
- Do you recognize Agile in any of these approaches?
- What can you tell us about Agile?
  - Which risks does it aim to manage?
  - When should it be applied – what is the value?

Lifecycle Planning

In your group,

Re-review the list of projects posted on D2L:

Lifecycle Planning - Projects

- Go back to the list of projects/scenarios on D2L
- **Decide** which scenario(s), if any could be managed using the approach you just summarized
- Which factors are you weighing in that decision?

In our first class I asked,
"What are the top factors that determine whether any software engineering effort will be successful?"

How does Lifecycle Planning factor into those answers?

Case Study 7-2

- Read the Case Study on page 159
  - What life cycle was chosen?
  - How was it chosen?
  - Was LifeCycle Planning properly done?

Lifecycle Strengths and Weaknesses

Table 7-1. Lifecycle Model Strengths and Weaknesses

Lifecycle Model Capability	Pure Waterfall	Code-and-Fix	Spiral	Modified Waterfalls	Evolutionary Prototyping
Works with poorly understood requirements	Poor	Poor	Excellent	Fair to excellent	Excellent
Works with poorly understood architecture	Poor	Poor	Excellent	Fair to excellent	Poor to fair
Produces highly reliable system	Excellent	Poor	Excellent	Excellent	Fair
Produces system with large growth envelope	Excellent	Poor to fair	Excellent	Excellent	Excellent
Manages risks	Poor	Poor	Excellent	Fair	Fair
Can be constrained to a predefined schedule	Fair	Poor	Fair	Fair	Poor
Has low overhead	Poor	Excellent	Fair	Excellent	Fair
Allows for midcourse corrections	Poor	Poor to excellent	Fair	Fair	Excellent
Provides customer with progress visibility	Poor	Fair	Excellent	Fair	Excellent
Provides management with progress visibility	Fair	Poor	Excellent	Fair to excellent	Fair
Requires little manager or developer sophistication	Fair	Excellent	'Poor	Poor to fair	Poor

The Reality:
Pure Waterfall Vs Pure Agile

Does anyone actually do Pure Waterfall or Pure Agile? Both have "workarounds" to compensate for their shortcomings.

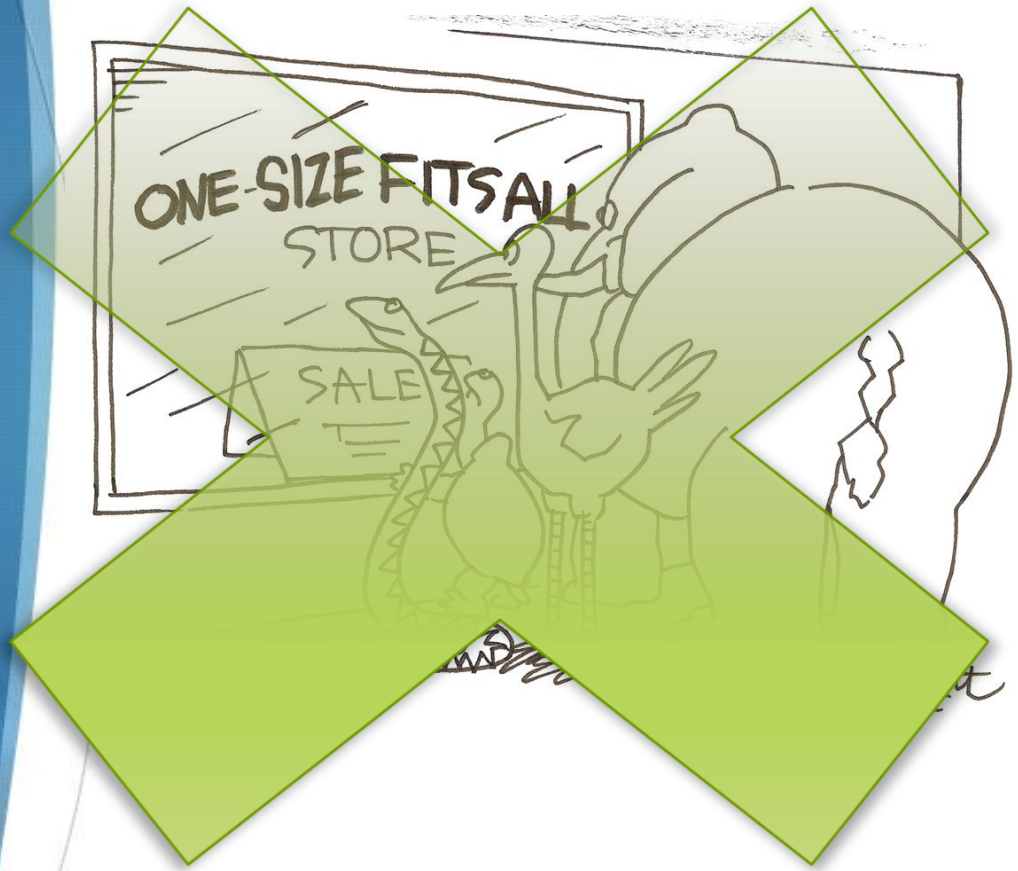
WATERFALL

- Modified Waterfall:
  - Overlapping Phases
  - Subprojects
  - Upfront Risk Reduction
- Staged Delivery
- Change Control

AGILE

- Upfront Roadmaps
- Upfront Design – Sprint 0
- Spikes for Timeboxed Investigations
- Scaled Agile Framework (SAFe)
- "Disciplined" Agile:
  - Requirements and Architecture are Upfront
  - Iterations are Design, Implementation and Unit Test
  - Test is Post-Iteration

Summary



Class Exercise

On D2L

For Next Week (week 6)

To Do before next class:

- Read McConnell, chapter 8: Estimation
- Read McConnell, chapter 9: Scheduling

Reminders:

- Assignment 1 due Feb 11th midnight (~1 week)
- Midterm on Feb 22nd, normal class room/time
  - I will do a (p)review next week

Comp8081

end of Week 5

Donna Turner