

The Software Lifecycle

PSU CS 300 Lecture 1-2

**Bart Massey
Assoc Prof Computer Science
Portland State University
<bart@cs.pdx.edu>**

How we build things

- **Figure out what is wanted**
- **Figure out how to build it**
- **Build it**
- **Make sure it was built properly and works right**

What we call these steps

- **Requirements**
- **Design**
- **Implementation**
- **Verification and Validation**
(**V&V**)

In SE, we say

- **The “Software Lifecycle”**
 - Requirements Gathering
 - Specification
 - Architecture
 - Detailed Design
 - Implementation
 - V&V

Lifecycle is a stupid name

- There's no “circle of life”
- In fact, dead software is *least* likely to be absorbed into “ecosystem”
- There's no “ecosystem”, either: nothing “alive”
- **Thinking, not analogies**

User Requirements (20%)

- **What is needed?**
- **“Customer” knows (maybe)**
- **Try to**
 - **identify customer**
 - **check understanding**
 - **document**

Requirements Spec (15%)

- **What will developer build?**
- **Organize, formalize requirements**
 - **functional**
 - **non-functional**
- **Identify (and limit) design constraints**

Architectural Design (15%)

- **General plan for SW**
 - Top-level decomposition
 - Description of interfaces
 - **Technological approach**
- **Start of design refinement**
- **Hard**

Detailed Design (15%)

- **Ideally, pseudocode**
- **Hierarchical decomposition complete**
- **All interfaces specified**
- **Enables any reasonably skilled implementor to succeed**

Implementation (20%)

- **Write the program**
- **Debug the program**
 - and the design
 - and the requirements
- **Traceability**

Verification & Validation (20%)

- **Plan first, verify throughout, validate last**
- **Not just testing**
 - testing
 - **inspection**
 - formal methods
- **Adversarial**

Phases are not process

- **“Waterfall Model” = just do phases in order given**
- **Better:**
 - **Iteration and increment**
 - **Spiral, prototyping**
 - **XP / open source**
- **Phases still occur**

Technical vs managerial

- **Lifecycle is about development = *project***
- **Management is about meta-development = *process*!**
- **Organizations etc are “products of management”**

Next up

- **Individual project management**