CS 5704: Software Engineering

Lifecycles

Dr. Pardha S. Pyla
Most important aspects

- risk
- people
- communication
- change
- complexity
- technology
A scenario

• Walk through of the steps for
  • Remodeling your kitchen
  • Remodeling your house
  • Remodeling someone else’s house
  • Building a whole housing colony
Two fundamental activities

Design -> specifications -> Implementation

constraints and problems
As complexity increases...

System analysis

Problem domain design

Software design

Software implementation

Software testing

Requirements → Constraints and problems → Requirements → Specifications

Test plan, criteria

Code modules → Errors, bugs

Major reconsiderations

Main feedback: design flaws, errors, modifications
One view: time

- Doing one activity at a time
- Doing part of each activity at a time
- Doing multiple activities simultaneously
- “Faking” upfront activities before committing resources
Another view: deliverables

- Standard, iron-clad deliverables for each activity
- Informal, loosely defined deliverables throughout
- Everything in between ...
Process and product

• “A Software Process is a framework for tasks that are required to produce high-quality software”

• Defines the approach to engineer software

• Includes technologies and tools (instances)

• Products are deliverables resulting due to the instantiation of the process

• Work products

• Software code
Possible to indirectly measure

- Process
  - CMMI *
  - Process metrics (cost, time, etc.)
- Product
  - Product metrics (LOC, function points, etc.)

* Open topic for presentation
Myths

- “SE is about creating unnecessary work products”
- “If we get behind schedule, we can add more programmers”
- “Let’s start coding, we can fill in details later”
- “Software is flexible, we can make changes anytime”
- “We have always done it that way”
Prescriptive Process Models

• Describe basic activities (analysis, design, etc.)
• Provides scaffolding to novices
• Incorporates control management mechanisms
• Emphasizes different key aspects we discussed
• Affords an engineering approach
The Waterfall Model

1. Design before code
2. Complete documentation
3. Prototype
4. Emphasize testing
5. Involve user
Waterfall Model discussion

• (Royce, 1970)
  • Context: chaos in the industry
  • What was actually proposed
  • Criticism (fair?)
  • What key concepts were identified?
  • What impact did the model have?
    • Whole organizational structures were defined!
Incremental process models

functionality

increment 1

increment 2

increment 3

delivery

time

delivery

delivery
Incremental Model discussion

- In some ways, precursor to the Agile methods
- What does this mean w.r.t. risk?
  - Importance of getting first increment right
  - Constraints down the road
- How does it impact users?
- Examples: OmniFocus on the Mac
The RAD Model

- Requirements
  - Analysis
    - Design
      - Implement
        - Deploy

Teams:
- Team 1
- Team 2
- Team 3

60-90 days
RAD Model discussion

- Emphasis on short development cycle
- Component based, parallel development
- What does it mean for communication?
- What does it mean for risk?
- Examples?
Evolutionary process models

- Prototyping
- What are they suitable for?
- What are the risks?
- How does this impact the users?
The Spiral Model

- Determine objectives, alternatives, constraints
- Plan next phases
- Evaluate alternatives, identify, resolve risks
- Develop, verify next-level product
- Cumulative cost

Plan next phase
Spiral Model discussion

- Biggest contribution: formalization of risk
- Specifications ~ risk ~ prototype ~ risk ~ ....
- What type of development efforts need this?
- Not many case studies on instantiating this model
Other models

- Concurrent Development Model
  - Controlled chaos?
- State-based, triggers in the development space
- Component-based Development Model
  - Holy grail of SE
- Formal Methods Model
  - Extremely accurate and mostly impractical