CSE 403
Software Engineering

Lecture 24
Review

Lecture 1 - Overview
- Course goal – gain an understanding that:
  - Developing a software product is not merely a matter of programming

Lecture 2 - Life cycle
- Software life cycle
  - Waterfalls and spirals
- Requirements
  - Gather and document the functions that the application should perform for the users in the users' language and from the users' perspective

Lecture 3 - Requirements
- Importance of up front work
- Types of requirements
  - Business Case
    - Why we want to build it
  - User
    - What it does
  - Functional
    - How developers are going to implement it
- Use Cases

Lecture 4 - User studies
- Understand application from users point of view
- Methodology
  - Observations
  - Interviews
  - Examination of artifacts
  - Focus groups
- Understand use case – not the implementation

Lecture 5 - Using requirements
- Who uses the requirements
  - Customer, Management, Marketing, Dev, Test
- Managing requirements
  - Change tracking
- Trade offs
  - Conciseness vs. Completeness
  - Formality vs. Comprehensibility
Lecture 6  Non-functional requirements

- Requirements beyond user interaction with the system
  - Performance
  - Safety
  - Security
  - Software quality attributes
- Domain has a strong influence on these

Lecture 7  How to fail at delivering software

- It's not just software . . .
- What is failure
- Problems can be covered up until it's too late
- Classic mistakes
  - People
  - Process
  - Product
  - Technology

Lecture 8  Risk Assessment

- What is risk
- Risks in different domains
- Sources of risk
  - Development
  - Integration
  - Testing
  - Deployment
- Risk mitigation

Lecture 9  Project teams

- Team structure
- Development roles
- Productivity differences
- What makes a team successful
- Team building

Lecture 10  Motivation

- Motivation is undoubtedly the single greatest influence on how well people perform
- Key factors
  - Achievement, Growth, Work itself
- Morale killers
  - Crushing motivation from the top

Lecture 11  QA (Ian King)

- Value of quality assurance early
- Establish QA requirements
- Process
  - Schedules
  - Build
  - Developer
  - Defect tracking
  - Release criteria
Lecture 12 Software decomposition
- Design principles
- Managing complexity
  - Divide and conquer
  - But where to divide?
- Cohesion vs. Coupling
- Language support

Lecture 12 ½ David Socha
- Urban Sim development approach
- Customer oriented
- Very short release cycle
- Emphasis incremental build
- Attention to work environment
  - Artifacts and atmosphere

Lecture 13 Urban sim discussion
- Quick release cycle
- Avoiding test
  - Devs test
  - Customer testing
- Code / debug cycle

Lecture 14 Design Patterns
- Expertise – higher level thought
- Design pattern
  - Commonly occurring pattern that can be adapted in many situations
  - Gang of Four – catalog of software design patterns

Lecture 15 Coding
- Many choices in how to do the same thing
- What criteria to use to make choices
- Coding standards
  - Layout, Commenting, Language usage
  - Naming issues

Lecture 16 Coding
- Code for comprehensibility
- Self documenting code
- Assertions
- Dealing with memory
Lecture 16 ½  John Socha

- Range of software experiences

Lecture 17  Coding

- Danger of bad function interfaces
  - Return of error conditions
- Code tuning
  - Out of fashion
  - Almost always favor clarity over efficiency
  - But be aware of
    - Speeding up the inner loop
    - Memory management
    - Resource usage
    - Avoiding recomputation

Lecture 18  Debugging

- Debugging techniques
  - Kernigan and Pike
- Common mistakes
  - Initialization / memory errors
  - Ignoring exceptions
  - Complicated interaction of systems

Lecture 19  Skip Walter

- Software Products
- Technology adoption lifecycle
- Product context
- Whole product concept

Lecture 20  The End game

- Release process
  - Release Candidate
  - RTM
  - Maintenance
- Influence of industry on process

Lecture 21  Test Strategy

- Ian King

- Elements of test strategy
  - Specification
  - Plan
  - Test architecture
  - Test case generation
  - Schedule
- Focus
  - The customer
  - The customer
  - The customer
Lecture 22 Ian King
Implementing Testing
- Phases of testing
- What makes a good tester
- Testability
- Black box vs. White box
- Test cases
- Manual vs. Automated
- Bug management

Lecture 23 Complexity theory!
- Syntactic measures of software quality are dubious
  - Mechanical simplification is not simplification
- The halting program
  - Cannot automatically verify software (in all cases)

Lecture 24 Review
- Requirements phase
  - Planning, teams, and management
  - Design and process
- Coding
- Shipping
- Review