

# Process Basics

## Object Oriented Analysis and Design

Aron Trauring

T++ Technical Skills Training Program

CUNY Institute for Software Design & Development (CISDD)

New York Software Industry Association (NYSIA)

November 29th, 2004

## Early Days

- The Fifties - Hardware dominates
- High level languages - Fortran, Cobol
- Small and efficient programs
- Compiler tricks

## The Sixties - The rise of software

- 1965 G. Moore's (Intel) Law - 'The density of chips doubles every year'
- 1968 Data General Nova - 32K \$8,000
- Software costs exceed hardware
- OS/360 + High level languages - programs portable and durable

# New Criteria for Successful Software Development

- A relatively low cost of initial development.
- Easily maintained.
- Portable to new hardware.
- Does the job the customer wants.

# Structured Programming, Analysis and Design

- Developing programs top-down (as opposed to bottom-up).
- Using a set of specific formal programming constructs (the go to was to be banished).
- Following some formal steps to decompose the larger problem.
- Human readable vs. Machine readable
- Software Engineering
- Methods Wars

- Guru Consultants - Edward Yourdon, Peter Coad, James Martin, and Tom DeMarco

## Fred Brooks - The Mythical Man Month

- PM of IBM s Operating System/360 (OS/360) in the early 1960's.
- Software development a human-centric process, not an engineering discipline.
- Why programming is hard to manage
- Why Projects Fail

## The Mythical Man Month Method (FourM)

- Reduce Complexity
- Enhance Communication
- Project Design - Conceptual Integrity
- Project Structure - The Surgical Team
- Project Implementation - Iterative and Incremental Development (IID)
- Project Communications — The Documentary Hypothesis
- Project Organization — Plan the Organization for Change

## SDLC — Project Phases

1. Requirements — What the Customer wants
2. Analysis — Understanding the Requirements — Functional Specification
3. Design — Software Architecture/Decomposition — Technical Specification
4. Development/Implementation — Writing the code — Programs
5. Unit Test — working unit
6. System Test — working system
7. Acceptance Test — Working application

## Eighties — New Directions

- SDLC a failure
- Projects grow more complex
- Search for new paradigms

## OOA/D — Key Concepts

- Objects basic building blocks
- Objects interact through exchanging messages
- Internal behavior is hidden
- Divide and Conquer
- Close connection between domain model and implementation

## OOA/D — UML

- Structured Analysis gurus picked up ideas of OO especially with DOD backing of ADA in 1983
- Notational and methods wars raged for fifteen years
- 1997 Object Management Group issued first UML standard
- ShortFoilheadized notation for modeling software systems
- A tool for improving communications
- UML is a notation not a method

## OOA/D — UP

- Three amigos — three gurus, Grady Booch, Ivar Jacobson, and Jim Rumbaugh
- Rational Systems — now owned by IBM
- RUP or UDP
- High ceremony tradition but influenced by agilists

## UP — Key Practices

- Develop in short time-boxed iterations
- Develop high-risk, high-value elements first
- Focus on customer value — Milestone goals
- Accommodate change early

## The Agile Manifesto — February, 2001

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

# Iterative Development

- Overall life-cycle is composed of multiple iterations
- Each iteration goes through all waterfall activities
- Iteration release at end of each cycle - stable, tested, integrated, partially complete

# Iterative Incremental Development

- Each iteration adds new features
- Number of iterations varies with project and method
- Not the same as prototyping
- IID is 100% FourM compatible
- All Agile methods are IID
- Agile <> IID
- Older IID methods (e.g. RUP) are adapting to/adopting Agile concepts

## XP — Historical Background

- Kent Beck and Ward Cunningham (Tektronix research)
- CRC cards
- Design Patterns (Smalltalk)
- Cunningham invented the Wiki
- C3 payroll project at Chrysler (Martin Fowler, Ron Jeffries)

## XP — Key Practices

- On-site customer as part of team
- Extreme avoidance of up-front design work
- Automated acceptance tests / test-driven development
- Pair programming