talentech

Software Development Life Cycle

Course Objectives

- Software Development Life Cycle (SDLC) Concepts
- Software Development Concepts
- SDLC Stages
- Teams involved
- Methodologies
  - Waterfall
  - Iteration
  - Agile
- Advantages and Disadvantages of Methodologies

INTRODUCTIONS

- Name
- Where From
- Education & Current Job
- Prior Experiences with SDLC

Software Basics

Understanding the Concept of Software

Basics of Software

- What is a Software?
  - Consisting of programs enable a computer to perform specific tasks
  - The programs and instructions that make the computer do some task, such as word process, manage databases, play games, etc.
  - Automation of Manual Work
- Name some Software?
  - MS Office, Adobe Photoshop
  - Windows, DOS, Oracle
  - USPS.com, Monster.com,

Basics of Software

- Why we need Software?
  - Fast – Perform tasks faster than human
  - Reliable – Accounting software are able to eliminate human errors, and provide reliable record keeping. Perform Complex Calculation.
  - Save Money – Less resources & efficient.
- How we build a software?
  - Programs are coded using programming languages and work as individual unit
  - The unit communicate and works together
  - Continuous development and enhancements of the units
  - Software developments follows defined life cycles

Types of Software

- Systems Software i.e. OS, Utilities such as Windows, Unix, Drivers for Printers
- Application Software i.e. programs that do real works for users such as word processors, DBMS.
- Desktop Application – MS Office, Adobe Photoshop
- Programming Software – Java, C++, C#, VB Scripts
- Client Server/ Distributed Application – Bank Teller system, ATM Machines
- Web Applications – USPS.com, Monster.Com, BankofAmerica.Com

Software Development Life Cycle (SDLC)

Intro of SDLC with the stages and teams involved

Software Dev Life Cycle - SDLC

- SDLC stands for
  - Software/ Systems Development Life Cycle
  - First, SDLC is a *Life Cycle*.
  - All systems have a life cycle or a series of stages they naturally undergo.
  - The number and name of the stages varies, but the primary stages are conception, *development*, Operation and Maintenance.
  - The **software development life cycle (SDLC)** therefore, refers to the development stage of the software's life cycle.

SDLC – Good or Bad?

- Major advantages
  - Control
  - Accountability
  - Error detection
- Major drawbacks
  - Relatively inflexible
  - Time-consuming and expensive
  - Discourages changes once user requirements are done

Software Development Stages

- Conception
- Requirement Gathering
- Systems Design
- Development
- Testing
- Implementation &Maintenance

Teams on the Stages

- Users/ System stakeholders
- Managers/ Directors
- System analysts
- System Architect
- Programmers
- QA/ Tester
- CM
- Help Desk/ Support

Conception – SDLC Stage

- Begins with the business problem (or opportunity) followed by the feasibility analysis.
- Feasibility study
  - Technical
  - Economic
  - Behavioral
  - Organizational
- Go/No-Go Decision

Teams Involved

- **Users** are employees from all functional areas and levels of the organization who interact with the system, either directly or indirectly.
- **System Stakeholders** are Personals who has direct interest to the system. These are are all people affected by changes in the information systems.
- **Directors and managers** manages the system development teams.

Testers Responsibilities - Conception

Software Testers are not active at this stage, but participate on meetings, discussions, and review documents.

Requirement Gathering

- A requirement for a computer system specifies what you want or need from that system.
- Defining requirements is the critical first step in the development life cycle. Requirements become foundation on which a project is built.
- Poor Requirements Cost Time & Money
  - 80% of maintenance efforts are spent on problems that arise from incomplete or undefined requirements.
- Types of Requirements
  - Business
  - User
  - Functional
  - Non-functional

Requirements – Cont'd

- Business requirements are based on the high level objectives of the company.
    - Example: of business requirements for an online insurance sales application is to gather all of the required information when selling a policy.
- User requirements are based on user goals or on what the user must be able to do with the product to achieve the business requirements of the company.
    - Example: USPS system should provide user to find a correct zip code based on the address provided.
- Functional requirements define what must the product do to enable users to accomplish their tasks.
    - Example: The software will prompt the user if the user id and password is incorrect.
- Non-Functional requirements outlines the constraints that must be considered while designing the system.

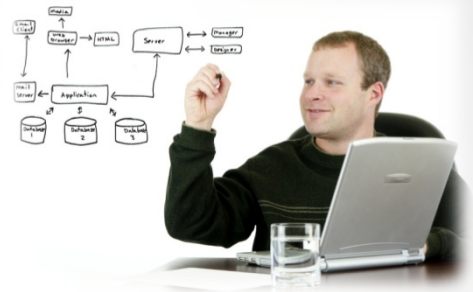
Testers Responsibilities - Requirements

QA Analysts/ Testers actively analyze requirements after the documents are ready by Business Analysts (BA).

Requirements are analyzed based on the following criteria:

- Clear
- Consistence, and
- Testable

System Design



- Describes how the system will accomplish this task.
- **Logical system design** states *what* the system will do, using abstract specifications.
- **Physical system design** states *how* the system will perform its functions, with actual physical specifications.
- Deliverable is the ***technical design*** that specifies:
  - System outputs, inputs, user interfaces;
  - Hardware, software, databases, telecommunications, personnel & procedures;
  - Blueprint of how these components are integrated.

Testers Responsibilities - Design

- Testers/ QA analysts is responsible to analyze and Review the design documents and Graphical User Interfaces (GUI)
- Some projects develop prototype, and Test Engineers may review the prototype to get familiar with the application and develop test cases.

Developments (Coding)

- Programming involves the translation of a system's design specification into computer code.
- Programming tools like compilers, interpreters, debuggers are used to generate the code
- Different high level programming languages like C, C++, Pascal, JAVA are user for coding.

Testers Responsibilities - Development

- Black box testers are not active on this stage rather trying to understand the challenges of specific modules
- White Box testers are active on Unit testing the code while the developments are on going.

Testing

- **Testing** check to see if the computer code will produce the expected and desired results under certain conditions.
- What is Testing?
  - Software testing is the process of planning, preparing, executing, and analyzing system establishing reasonably bug free application.
  - Two main concerns about the quality of software are as follows
    - ***Verification*** – Building a system that conforms to requirements
    - ***Validation*** – Building a system that accomplishes these requirements

Testers Responsibilities – Test Phase

QA Analyst/ Test Engineers are very active at this stage and 80% of testers responsibilities fall under this stage. The following levels and types of testing are done on this stage:

- Integration
- System
- User Acceptance
- Functional & Regression
- Load/ Performance

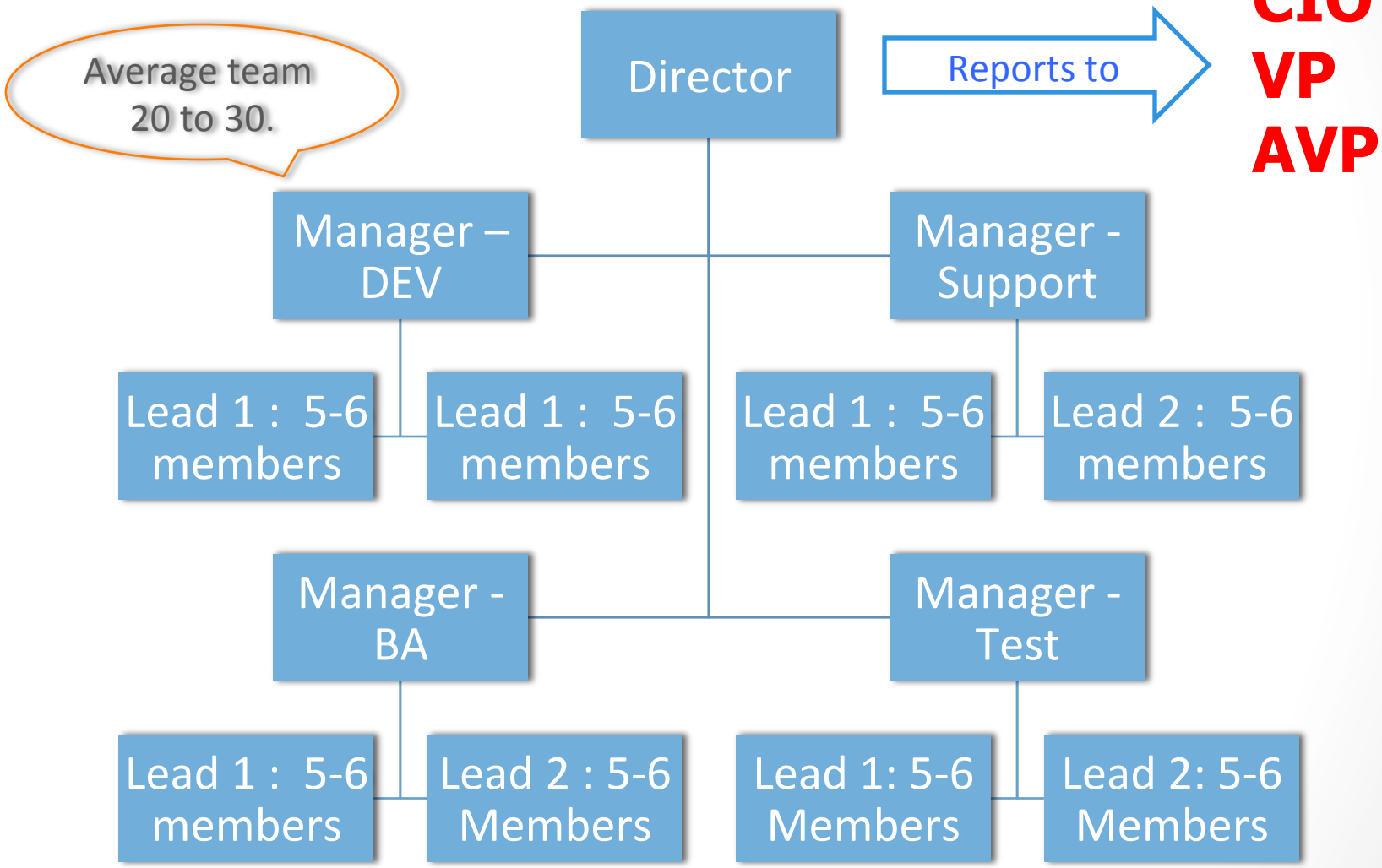
Implementation & Maintenance

- **Implementation** or deployment is the process of converting from the old system to the new system.
- **Audits** are performed to assess the system's capabilities and to determine if it is being used correctly.
- Systems need several types of maintenance.
  - **Debugging**: A process that continues throughout the life of the system.
  - **Updating**: Updating the system to accommodate changes in business conditions.
  - **Maintenance**: That adds new functionally to the system – adding new features to the existing system without disturbing its operation.

Testers Responsibilities - Implementation

- Testers perform regression testing of defects found and impacted features of the defects
- There are reduced testing activities at this stage

Org Chart – SDLC Team



SDLC Teams (Cont'd)

- **System analysts** are Information professionals who specializes interacting to business users and technical users to identify requirements.
- **Systems Architects** are designer experts on application/ system design.
- **Programmers** are IS professionals who modify existing computer programs or write new computer programs to satisfy user requirements.

SDLC Teams (Cont'd)

- **QA Tester/ Analyst** performs software quality assurance and testing in all stages starting from requirements to implementation.
- **Configuration Managers** are responsible for managing software environments and make sure that the established processes are followed among software development teams.
- **Help Desk/ Support** provides trouble shooting for all users in software development life cycle. Supports includes installation, PC troubleshooting, network maintenance. Etc.

Summary

SDLC Phase	What	Who	Docs
Concepts	Feasibility Studies	Owner/ Managers	Meeting notes/ Feasibly studies
Requirements	Requirement gathering	BA	BRD/URD/FRD
Design	System Design	System Architect	Design Docs
Development	Coding	Programmer	Code design
Testing	Validation	QA Test Analysis	Test Case/ Defect/ Reports
Implementation	Release to user	CM	User Guides

Methodologies

SDLC VS Methodologies

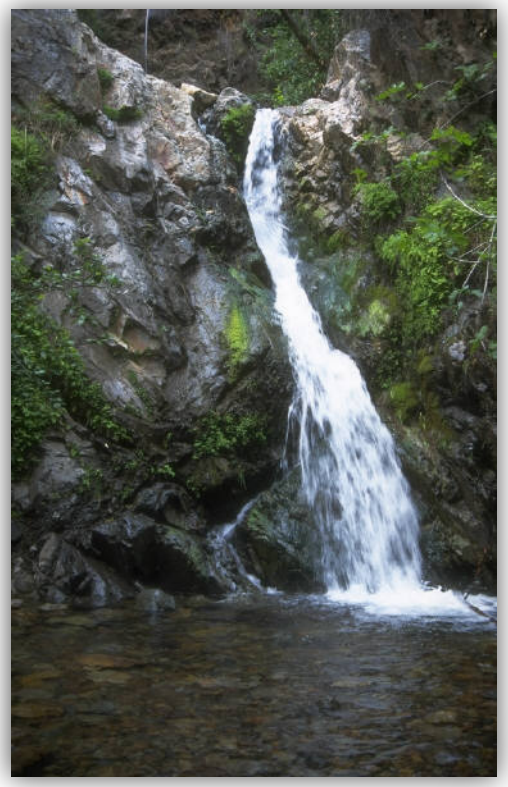
- **SDLC** refers to a stage all systems *naturally* undergo, a methodology refers to an approach *invented by humans* to manage the events naturally occurring in the SDLC.
- A **methodology** is, in simple terms, a set of steps, guidelines, activities and/or principles to follow in a particular situation.
  - Most methodologies are comprehensive, multi-step approaches to systems development
  - There are many methodologies out there.
  - You may come up with a methodology.

SDLC Methodologies

- Waterfall
- Iterative
- Agile

Waterfall

- The waterfall model is an activity-centered life cycle model that prescribes a sequential execution of a subset of the development processes and management processes.
- Every stage must be completed before going forward to next stage. For example the requirement activities are all completed before the system design activity starts.
- The goal is to never turn back once an activity is completed



Waterfall Model – When & Why

- **When to Use The Waterfall Model?**
  - A set of high quality, stable user requirements exist
  - The user require all complete system at once
  - Previous experience of building similar systems exist
  - The duration of the project is two years or less
- **Advantages:**
  - Easy to manage because it allows for departmentalization and managerial control.
  - A schedule can be set with deadlines for each stage of development delivered on time.
  - Each phase of development proceeds in strict order, without any overlapping or iterative steps.

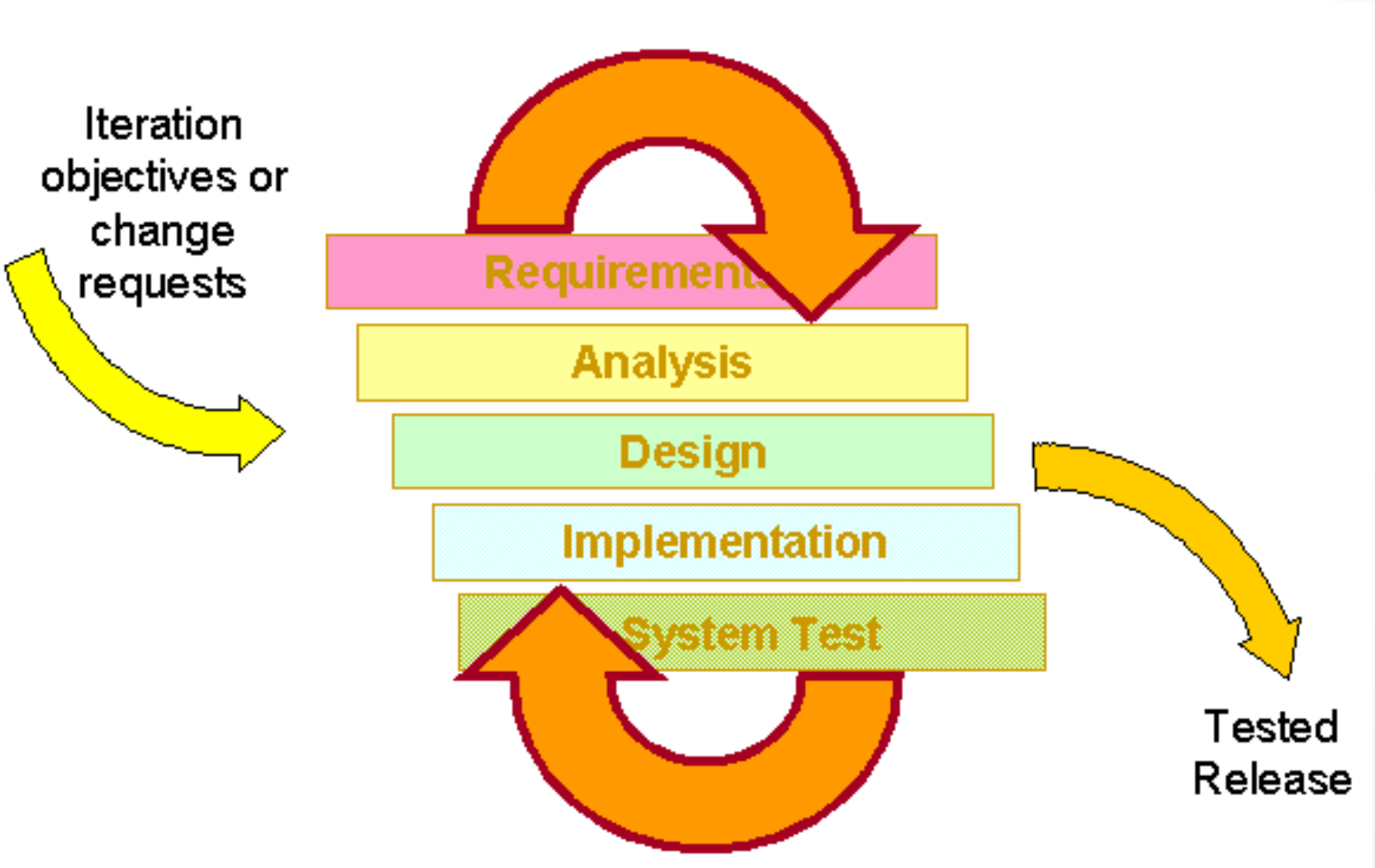
Waterfall Model – Cont'd

- **Disadvantages:**
  - It is not always possible to gather requirements all at the beginning in order to properly design the system, not all requirements are received at once,
  - The requirements from customer goes on getting added to the list even after the end of Requirement Gathering phase, this affects the system development process and its success in negative aspects.
  - The problems with one phase are never solved completely during that phase and in fact many problems regarding a particular phase arise after the phase is signed off.
  - The project is not partitioned in phases in flexible way.

Iterative Methodology

- It is an approach to building software in which the overall life cycle is composed of several iterations in sequence
- Each iteration is a self-contained mini project composed of activities such as requirement analysis, design, programming, and test
- The goal for the end of iteration is an iteration release; a stable integrated, and tested partially complete system

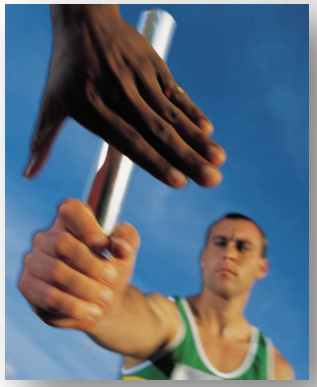
Iterative Model



Advantage – Iterative Model

- The design phase goes much faster
- Coding and testing goes much faster
- The client gets into production in less than 3 months

Agile



- Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
- It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change.
- Agile methods break tasks into small increments with minimal planning and do not directly involve long-term planning.
- Iterations are short time frames that typically last from one to four weeks.

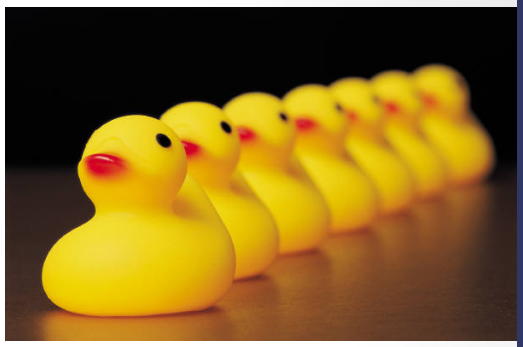
Agile

- Each iteration involves a team working through a full software development cycle, including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders. This minimizes overall risk and allows the project to adapt to changes quickly.
- An iteration might not add enough functionality to warrant a market release, but the goal is to have an available release at the end of each iteration. Multiple iterations might be required to release a product or new features.
- Team composition in an agile project is usually cross-functional and self-organizing, without consideration for any existing corporate hierarchy or the corporate roles of team members. Team members normally take responsibility for tasks that deliver the functionality an iteration requires. They decide individually how to meet an iteration's requirements.

Sprint - Agile

- A sprint is the basic unit of development in Agile. Sprints last between one week and one month, and are a "time boxed" (i.e. restricted to a specific duration) effort of a constant length.
- Each sprint is preceded by a planning meeting, where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made, and followed by a review or retrospective meeting, where the progress is reviewed and lessons for the next sprint are identified.
- During each sprint, the team creates finished portions of a product. The set of features that go into a sprint come from the product *backlog*, which is a prioritized list of requirements.

Daily Scrum - Agile



- Agile methods emphasize real time communication, preferably face-to-face, over written documents
- Most agile implementations use a routine and formal daily face-to-face communication among team members called “Daily Scrum”
- Scrum is facilitated by a Scrum Master who is accountable to deliver the sprint goal/deliverables. The Scrum Master ensures that the Scrum process is used as intended
- The objective of Daily Scrum is to discuss followings for each team members
  - What is the status of current tasks at hand?
  - What will the team be working today and tomorrow?
  - Any major issues that requires escalation?

Testing in Agile

- One of the differences between agile and traditional testing methods, such as the waterfall or Iterative, is that testing of the software is conducted at different points during the software development lifecycle.
- Traditional methods (waterfall, Iterative) advocate testing after development, either after complete system development (waterfall) or after increments in a more iterative process (Iterative).
- Agile advocates techniques such as Test Driven Development(TDD) where the tests are written before the code or Behavior Driven Development where the scenarios are written before the code that can verify whether the code meets the required functionality and/or quality standards

Criticism - Agile

- One common criticism of agile software development methods is that it is developer-centric rather than user-centric. Agile software development focuses on processes for getting requirements and developing code and does not focus on product design.
- Agile methodologies can also be inefficient in large organizations and certain types of projects. Agile methods seem best for developmental and non-sequential projects. Many organizations believe that agile methodologies are too extreme, and adopt a hybrid approach that mixes elements of agile and plan-driven approaches.

Now What...

What's Next?

Homework

- Prepare 3 - 5 minutes presentation on the following topics
  - Software Development Stages
  - Software Development teams
  - Waterfall
  - Iterative
  - Agile
- Discuss the software development methodology you prefer, and explain why?

Questions drive results.