RUP & Agile (Scrum)
Waterfall

Traditional way to build systems

Sequential

- **detailed planning**
  - problem is identified, documented, designed
  - implementation tasks are identified, scoped and scheduled
  - approvals & revisions

- development cycle

- testing cycle

- bug fixing cycle
Waterfall

Strengths

• logical
  – requires preparation before execution

• organized
  – documented,
  – planned
  – deviations are exceptions and are tracked
Waterfall

Weakness

• not very flexible
  - good ideas need to be identified upfront
  - but what if I get an idea midway through development?
  - “a great idea late in the release cycle is not a gift, it’s a threat”

• documentation heavy
  - abstract
    need to protect data
    use encryption for storage
Process

- configurable
  - no single process is suitable for all software development.
  - adapts to small & large development teams
- documentation
  - model based artifacts
  - UML
Building blocks

• roles (who)
  – responsibilities

• tasks (how)
  – unit of work
  – result oriented – should be useful

• work products (what)
  – resultant product
Life-cycle Phases

- **four phases**
  - inception, elaboration, construction, transition

- **characteristics**
  - sequential in nature
    - hmm... sounds like waterfall methodology
  - each phase focuses on a
    - key objective
    - milestone delivery
RUP – Life-cycle Phases

Inception
• vision document
  – scope the system
  – identify major players
  – risk, cost etc

Elaboration
• risk identification
• problem domain
• analysis & architecture

Construction
• build the software
• can be broken down into iterations

Transition
• transition from development to production
RUP Engineering Disciplines

**Business modelling**
- domain understanding

**Requirements**
- vision document & use cases

**Analysis & Design**
- blueprint for system realization

**Implementation**
- develop components

**Test**
- testing throughout the project

**Deployment**
- product releases
- software delivery
RUP

Iterative Development
Business value is delivered incrementally in time-boxed cross-discipline iterations.

<table>
<thead>
<tr>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>E1</td>
<td>E2</td>
<td>C1</td>
</tr>
</tbody>
</table>

- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

Time
RUP Best Practises

Develop iteratively

- not possible to
  - define the problem upfront
  - design the entire solution
- each iteration ends with a release

Manage requirements

- use case to capture functional requirements
- should be traceable
RUP Best Practices

Use components

Model visually

- different models to communicate
  - different aspects
  - with different stakeholders
  - UML
RUP Best Practises

Verify quality

• review
  – functional requirements
  – non-functional requirements

• should be part of the process

Control changes

• continuous integration
Scrum

Principles

• building working software that people can get their hands on quickly
• cross functional teams empowered to make decisions
• rapid iteration with continuous customer input
Scrum in a Nutshell

Framework

- iterative & incremental

Sprint

- development done in sprints (cycles)
- sprints are time-boxed
  - end after one month no matter what
- deliverables are static for each sprint
- progress reviewed at the end of each sprint
- goal: deliver working product
Inspect & Adapt

Intent

- “development inevitably involves learning, innovation and surprises”

Recipe

- take a short step
- inspect result & practise
- adapt if required
- repeat forever
Meet the Players

Product Owner

- **objective is to maximize ROI**
  - has profit & loss responsibility
- **identify product features (product backlog)**
- **define feature prioritization**
  - satisfy key stakeholders
  - alignment with other strategic objectives
  - risk identification
- **actively interact with the team**
Meet the Players

The Team (pigs)

• builds the product
• cross functional
  – analysis, development, testing, interface design, documentation, database design etc..
• self organizing
• small in size (roughly seven)
• can be feature specific
Meet the Players

Scrum Master (Mr. Scrum)

- acts as a Scrum educator and a facilitator
  - does whatever in his power to help the team and product owner be successful
- should not be same as the product owner
  - conflict of interest
Define sprint

- identify the items from "release backlog"
- assign weight to each item
  - product owner – based on value
  - team – based on effort
- lock commitment
Scrum in Action

Daily scrum

- 15 mins every day
- Monitor progress by answering
  - What have you done since yesterday?
  - What are you planning to do by today?
  - Do you have any problems preventing you from accomplishing your goal?
Scrum in Action

Monitoring progress

- **burndown chart**
  - reach zero effort by the last day of sprint

http://www.xqual.com/resources/images/scrum_burndown_chart.gif
Sprint Review Meeting

- review the work that was
  - completed
  - not completed

- Inspect & adapt
RUP or Agile

Similarities

- iterative
- division of work
- continuous testing
RUP or Agile

Differences

- Management style
- RUP is predictive, agile is adaptive
- Customer interaction
- Agile requires a seasoned team
- Knowledge sharing