Information about Life Cycles

• Serial Life Cycle: Waterfall or Phase-Gate
• Iterative Life Cycle: Spiral Evolutionary Prototyping Unified Process
• Incremental Life Cycle: Staged Delivery Design to Schedule
• Agile Life Cycles
Serial Life-Cycle

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Analysis</th>
<th>Design</th>
<th>Code</th>
<th>Integration</th>
<th>Test</th>
</tr>
</thead>
</table>

- Five stages
- Management review at the end of each stage
- Baseline products produced

Deliverable-based Planning

- Risks
  - Architectural risk
  - Testing risk
  - Schedule risk
Verification (testing the product as you go to make sure it will work)
Validation (testing the process to make sure development is correct)

Replan with a Serial Life Cycle

Rolling Wave Scheduling
– Continuous – few weeks long
– Complete one week, add a week of detail
– Replan each week with increased knowledge
# Iterative Life Cycle

|--------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------|-------------|------|

## Spiral

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Design and implement initial Prototype: Obtain customer feedback.</th>
<th>Refine Prototype. Obtain feedback</th>
<th>Refine Prototype. Obtain feedback</th>
<th>Refine Prototype. Obtain feedback</th>
<th>...</th>
<th>Complete Prototype. including integration and final testing.</th>
</tr>
</thead>
</table>

## Evolutionary Prototyping

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Timeboxed iteration, developing higher risk &amp; value pieces first</th>
<th>Timeboxed iteration, developing higher risk &amp; value pieces first</th>
<th>Timeboxed iteration, developing higher risk &amp; value pieces first</th>
<th>...</th>
<th>Final Development iteration</th>
<th>Final Integration</th>
<th>Test</th>
</tr>
</thead>
</table>

## Unified Process

*Manage It! Your Guide to Modern, Pragmatic Project Management. Johanna Rothman*
Iterative Life Cycle

Benefits:
• Helps with frequently changing requirements
• Technical risks minimized
• Better idea of progress
• Less risky moving members in and out

Risks:
• Schedule (finishing can be difficult)
• Cost risk (go after riskiest & not most valued)
• Architectural risk
• Developed feature might change
Agile

One iteration

System demo after each iteration

Start Iteration

Build at least daily

Iteration ends
Software is releasable

Need to know how to break things apart!
How Agile Helps

• Schedule risk
• Project team changes (maybe)
• Requirements change (maybe)
• Cost risk (reordering of iterations possible)
Where Agile breaks down

• Much, much more discipline is required… and if it doesn’t happen…
• Much more feedback is needed…

“If you or your management are not consistent with what you say or they say and do, team members will *flip the bozo* bit on you and eventually leave.”
Flip the Bozo bit 😊

Problem: There’s an idiot on your team (one who never contributes anything intelligent).

- Work needs to get done
- You cannot fire him or her
- You cannot get the him or her transferred
- You cannot physically assault him or her

Solution: Set their “Bozo bit” to TRUE

Everything he or she says can be safely ignored.

Use the time when they are speaking to plan what you are going to say or do next.