Choosing a Software Process Model

Comparing Process Models

Category	Characteristics	
Code & Fix	• Code used for analysis/problem solving • Small amount of code	• One developer • Thrown away after results obtained
Waterfall	• Requirements well defined • Predictable development • Known environment • Experienced development team	• Enhancing/upgrading existing system • No technological risks • Reasonable budget/schedule • Interfaces established/unchanging
Evolutionary (or Incremental Build) Model	• Most requirements well defined • Budget/schedule constraints • Some technology risks • Integration with other systems/ subsystems	• Some evolving requirements • Interfaces not well defined • New system, never done before • COTS and/or reusable software not available
Spiral Model	• Very large software system • Long schedule (5 to 10 years)	• Many risk areas • Emphasis on system reliability
COTS Integration	• Requirements can be satisfied by COTS products • Numerous functional capabilities in commercial	• Government furnished components • Intended to be inexpensive solution • Often assumed “simple” solution
Agile Development	• Requirements unstable • Unpredictable development • Unknown environment • Minimal planning	• Anticipate Requirements Volatility • People-oriented • Highly Collaborative • Focus on Construction

General Capabilities

Lifecycle Model Capability	Code and Fix	Waterfall	Evolutionary	Spiral	COTS Integration	Agile Development
Works with ambiguous requirements	Poor	Poor	Fair to Excellent	Excellent	Excellent	Excellent
Works with ambiguous architecture	Poor	Poor	Poor	Excellent	Poor to Excellent	Excellent
Produces highly reliable system	Poor	Excellent	Fair to Excellent	Excellent	Poor to Excellent	Poor to Excellent
Produces system with large growth potential	Poor	Excellent	Excellent	Excellent	N/A	Fair
Manages risks	Poor	Poor	Fair	Excellent	N/A	Fair
Can be constrained to a predefined schedule	Poor	Fair	Fair	Fair	Excellent	Poor
Minimal management and technical oversight needed to use the model	Excellent	Poor	Fair	Fair	Excellent	Fair
Allows for midcourse corrections	Poor to Excellent	Poor	Fair to Excellent	Fair	Poor	Excellent
Provides customer with progress visibility	Fair	Poor	Excellent	Excellent	N/A	Excellent
Provides management with progress visibility	Poor	Fair	Excellent	Excellent	N/A	Excellent
Requires little manager or developer sophistication	Excellent	Fair	Fair	Poor	Fair	Fair
Works with ambiguous requirements	Poor	Poor	Fair to Excellent	Excellent	Excellent	Excellent