**SDLC Modules**

**Incremental Development Pros and Cons.**

Pros:

* The cost of accommodating changing customer requirements is reduced.
* Easier to get customer feedback on development.
* More rapid delivery and deployment of useful software to the customer.

Cons:

* Process is not visible, so management has a hard time measuring progress.
* System structure degrades as new increments are added.

**Scrum.**

Key characteristics of sprints in scrum:

* Fixed length, normally 2-4 weeks
* The starting point for planning is the product backlog, which is the list of work to be done on the project.
* The selection phase involves all of the project team who work with the customer to select the functionality to be developed in the sprint.
* The team organizes themselves to develop the software through short daily meetings to review progress and possibly reprioritize work.
* At the end of the sprint, the work is reviewed and presented to stakeholders before beginning the next sprint cycle.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Pros | Cons | When (to use) |
| Waterfall | * Easy to understand, easy to use. * Provide structure to inexperienced staff. * Milestones are well understood. * Sets requirements stability. * Good for management control (plan, staff, track). * Works well when quality is more important than cost or schedule. | * All requirements must be known upfront. * Deliverables created for each phase are considered frozen – * inhibits flexibility. * Can give a false impression of progress. * Does not reflect problem-solving nature of software development - iterations of phases. * Integration is one big bang at the end. * Little opportunity for customer to preview the system (until it may be too late). | * Requirements are very well known. * Product definition is stable. * Technology is understood. * New version of an existing product * Porting an existing product to a new platform |
| V Model | * Emphasize planning for verification and validation of the product in early stages of product development. * Each deliverable must be testable. * Project management can track progress by milestones. * Easy to use. | * Does not easily handle concurrent events. * Does not handle iterations or phases. * Does not easily handle dynamic changes in requirements. * Does not contain risk analysis activities. | * Excellent choice for systems requiring high reliability –hospital patient control applications. * All requirements are known up-front. * When it can be modified to handle changing requirements beyond analysis phase. * Solution and technology are known. |
| Structured Evolutionary  Prototyping | * Customers can “see” the system requirements as they are being gathered. * Developers learn from customers. * A more accurate product. * Unexpected requirements accommodated. * Allows for flexible design and development. * Steady, visible signs of progress produced. * Interaction with the prototype stimulates awareness of additional needed functionality. | * Tendency to abandon structured program development for “code-and-fix” development. * Bad reputation for “quick-and-dirty” methods. * Overall maintainability may be overlooked. * The customer may want the prototype delivered. * Process may continue forever (scope creep). | * Requirements are unstable or have to be clarified. * As the requirements clarification stage of a waterfall model * Develop user interfaces. * Short-lived demonstrations * New, original development * With the analysis and design portions of object-oriented development |